Lower Bounds for Node Search Number on Grid-like Graphs

By

Robert Bramwell Warren
B.Sc., Memorial University, 2003

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

University of Victoria

Supervisor: Dr. J. Ellis

ABSTRACT

One method to find the node search number of a graph is to prove identical upper and lower bounds. In four types of grid-like graphs, (h, w)-grids, cylinders, orb webs, and walls, upper bounds are easy to see. However, for tori, the upper bounds are less obvious, requiring two different search strategies. In all cases the lower bounds are not obvious and previously unproven. For these five classes of graphs we develop several techniques for proving lower bounds by taking advantage of the fact that recontamination does help. We observe that in the five classes of graphs we examine, the node search number can be expressed as a function of the height and width of the graph.

Contents

List of Figures

Chapter 1

Introduction

Finding the boundary between tractable and intractable problems, in the field of computational complexity, is an important area of computer science. Even with the incredible power and resources possessed by today's computers, intractable problems remain, and are likely to remain, nearly impossible to solve. Thus, understanding which problems are intractable is important to understanding the limitations of modern computer science.

In this paper we present a small step in the field of computational complexity. We narrow our focus to exploring the complexity of the graph problem of node search number. Node search number is best described as a game played out on a graph. The object of the game is to capture an intruder moving around the graph by deploying guards at the various vertices of the graph. The node search number is the minimum number of guards needed to guarantee the capture of the intruder. In the general case, computing node search number is provably NP-hard [Len81] and hence is believed to be intractable. However, for some graph families, node search number has been found to be tractable. It is the boundary between those graphs for which node search number is tractable and those for which node search number is intractable that we will explore. In particular we are interested in some families of grid-like graphs, all of which are planar, except for the tori.

Currently, the dividing line between tractable and intractable for node search number on planar graphs is between outerplanar graphs, which are known to be in P [BK96] and planar graphs of maximum degree 3, which are provably NP-complete [MS88].

Node search number is an interesting problem to analyze because it is equivalent to a number of other graph problems. Perhaps the most important of these other problems is pathwidth. Some problems that are NP-hard are solvable in polynomial time for graphs with fixed pathwidth, for example, Network Reliability and 2-Edge-Connected Reliability [LMC00].

Aside from the general importance of node search number, there are several applications. In computational biology, an important problem is the detection of false negatives during the physical mapping of DNA. While this problem is known to be hard, one approach is to generate a map of minimum width. Minimizing the map's width is equivalent to minimizing the node search number [GGKS95].

In VLSI design, gate matrix layout is a technique for efficient chip design. Node search number is equivalent to gate matrix layout [LL80].

In natural language processing, words can be modeled as vertices in a graph, and where edges represent the semantic relationship between words. In order to communicate, words must be moved from long-term memory into short-term memory. This move is facilitated by a finite stack-like structure called a *shack*. In order to move a vertex from the shack into short-term memory, all vertices connected to it must either already be in short-term memory, or they must be in the shack (i.e., they cannot be in long-term memory). As it turns out, the node search number of the graph can be used to determine the demand on the shack.[KT92].

1.1 Background

Node search number, in the general case, has been proven to be NP-complete[Len81]. As a result, most research has focused on node search number when restricted to certain classes of graphs. The first restricted class of graphs to be studied was planar graphs. Node search number was proven to be NP-complete for planar graphs even when further restricted to planar graphs of maximum degree 3 [MS88]. In 1991 it was shown that, for n-dimensional grids, finding the node search number is in P [BL91]. In 1993, it was proven that node search number on cographs is in P[BM93] and node search number on chordal graphs is NP-complete [Gus93]. In 1994, Ellis, Sudborough, and Turner proved node search number is linear time for trees [EST94]. In 1995, it was discovered that node search number for permutation graphs is in P [BKK95] and node search number for bipartite graphs is NP-complete [GGKS95]. A year later, in 1996, it was shown that node search number for outerplanar graphs is also in P but the algorithm runs in $O(n^{11})$ [BK96]. Node search number for grid graphs and for unit disk graphs was proven to be NP-complete in 2001 [DPPS01]. More recently Ellis and Markov proved that the node search number of unicyclic graphs is in $O(n \log n)$[EM04].

Of course, with node search number in the general case and for many restricted cases being NP-complete, there has also been some exploration of approximation algorithms. Approximation algorithms are algorithms that produce the result in a polynomial amount of time but with a margin of error ε, such that the result x is $opt \leq x \leq \varepsilon \cdot opt$, where opt is the optimal result. So far the best approximation algorithm is for outerplanar graphs, which produces a result that is no worse than twice optimal [BF02]. An approximation does, however, exist for both the general case and for planar graphs. For the general case the result is no worse than $O(\log^2 n)$ times optimal [BGHK95]. For planar graphs the result is slightly better yielding a result that is $O(\log n)$ times optimal [BGHK95].

1.2 Results

Grids have a regular structure which makes it easy to develop some good searching strategies. However, finding a good strategy does not disprove the existence of an even better strategy. An upper bound on the node search number on grids may be obvious, but to prove that this is in fact the node search number it must also be shown that this is a lower bound.

In this thesis we find the lower bounds of (h, w)-grids and cylinders. We extend cylinders to cover some interesting planar variants which we call orb webs and walls. Finally, we explore tori, the only non-planar graph that we examine.

While proving the lower bounds on tori, an additional discovery was made. For tori where the height and the width are the same, the obvious upper bound is not the best upper bound. The strategy that achieves this better upper bound is complicated and not obvious.

Chapter 2

Common Definitions

2.1 Graph Theory

A *simple undirected graph* $G = (V, E)$ is an ordered pair consisting of two disjoint sets: a non-empty finite set V whose elements are called *vertices* and a set $E \subseteq \{\{x, y\} \mid x, y \in V \text{ and } x \neq y\}$ of unordered pairs called *edges.* We consider only simple undirected graphs so, henceforth, whenever we use the term "graph" we are referring to simple undirected graphs. As a shorthand, we will use $V(G)$ and $E(G)$ to refer to the set of vertices and edges in G respectively. If $u, v \in V(G)$ and $e = \{u, v\} \in E(G)$ then we say that u and v are *the endpoints*, that they are *adjacent* to each other, and that they are *incident* with e or that e is incident with u or v. The set of vertices adjacent to u is denoted by $\text{adj}(u)$. The *degree* of u is $\deg(u) = |\text{adj}(u)|$.

If $G = (V, E)$ is a graph, then any graph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, is a *subgraph of* G, which we denote by $G' \subseteq G$. If $V' \subset V$ or $V' = V$ but $E' \subset E$ then we call G' a *proper subgraph of* G, denoted by $G' \subset G$. If V'' is a vertex set such that $V'' \subseteq V$, then the subgraph $G'' = (V'', E'')$ where $E'' = \{\{u, v\} \in E \mid u, v \in V''\}$ is the subgraph *induced* by V''.

Let $G = (V, E)$ be a graph and $G' = (V', E')$ be a subgraph of G. To *delete*

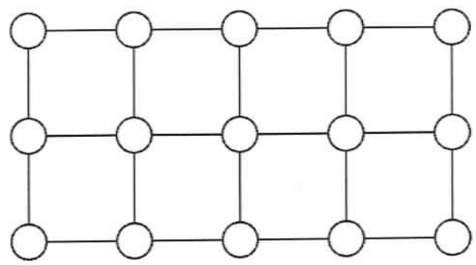


Figure 2.1: A (h, w)-grid graph, where $h = 3$ and $w = 5$

G' *from* G means to transform G into $G'' = (V'', E'')$ where $V'' = V \setminus V'$ and $E'' = \{\{u, v\} \mid \{u, v\} \in E \text{ and } \forall\{x, y\} \in E', \{x, y\} \cap \{u, v\} = \emptyset\}$. We denote this as $G'' = G - G'$. To *remove an edge e from G* means to transform G into $G^3 = (V, E \setminus e)$. We write $G^3 = G - e$. Let $u \in V$ and $E_u = \{\{u, z\} \in E \mid z \in \text{adj}(u)\}$. We *delete the vertex u* by transforming G into $G^4 = (V \setminus \{u\}, E \setminus E_u)$. This is written as $G^4 = G - u$.

A *path* from u_1 to u_n is a sequence $p = u_1 e_1 u_2 e_2 \ldots e_n - 1 u_n$ of alternating vertices and edges such that for $1 \leq i < n$, e_i is incident with u_i and u_{i+1}.

A graph G is *connected* if there is a path between any two vertices. If G is not connected, then the maximal connected subgraphs of G are called *the components of G*. In a connected graph, any vertex whose deletion creates more than one component is referred to as a *cut vertex.*

A graph G is planar if it can be drawn in the plane without graph edges crossing. In this paper we will look at several specific types of planar graphs: grid graphs, cylinders, orb web graphs and walls. We will also look at one non-planar graph, the torus.

2.1.1 (h, w)-Grid Graphs

A (h, w)-*grid graph* of height h and width w is the graph comprising the vertex set $\{(x, y) \mid 0 \leq x < h, 0 \leq y < w\}$ and the edge set $\{\{(u, v), (x, y)\} \mid |u - x| + |v - y| = 1\}$.

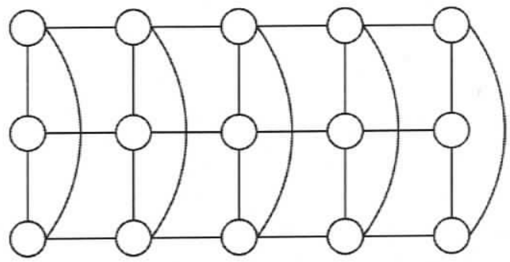


Figure 2.2: A cylinder graph, where $h = 3$ and $w = 5$

Grids are defined only for $h \geq 3$ and $w \geq 3$. See Figure 2.1.

We define a *row* to be the vertex set $\{\{i, y\} \mid 0 \leq y < w\}$ where $0 \leq i < h$. Similarly we define a *column* to be a vertex set $\{(x, i) \mid 0 \leq x < h\}$ where $0 \leq i < w$. If there exists a $u \in R$ and $v \in R'$ where R and R' are different rows and an edge $\{u, v\}$ then we say R and R' are *adjacent.* While rows and columns are collections of vertices, it is occasionally useful to speak of row or column edges. This refers to any edge which consists of only vertices in the same row or column.

Any row where $i = 0$ or $i = h - 1$ or any column where $i = 0$ or $i = w - 1$ is said to be a *side.* Note that there are four sides on a (h, w)-grid graph. If S is a side, and $v \in S$ then we say v is *peripheral.* If S_1 and S_2 are sides and $v \in S_1$ and $v \in S_2$ we say v is a *corner.* If there is a corner vertex v between two sides S_1 and S_2 we say that S_1 and S_2 are *adjacent.* Otherwise, we say S_1 and S_2 are *opposite.*

It is often helpful to speak of the *top* of the graph, referring to the row where $i = 0$. The *bottom* of the graph is a reference to the row where $i = h - 1$. The *left side* of a grid is the column where $i = 0$, while the *right side* of a grid is the column where $i = w - 1$. The *center* of a graph is the vertex $(\lfloor h/2 \rfloor, \lfloor w/2 \rfloor)$.

2.1.2 Cylinder Graphs

We picture *cylinder graphs* or *cylinders* as supergraphs of (h, w)-grid graphs with wrap-around edges in the vertical dimension. The vertex set for cylinders is the same as it was for (h, w)-grid graphs, $\{(x, y) \mid 0 \leq x < h, 0 \leq y < w\}$, but the edge set is now $\{\{(u, v), (x, y)\} \mid |u - x| + |v - y| = 1\} \cup \{(0, v), (h - 1, v)\}$. See Figure 2.2. We define a *row* and *column* and *peripheral vertex* as they were defined for (h, w)-grids. The definition for a *side* has, however, changed to any column where $i = 0$ or $i = w - 1$. Note that there are only two sides on a cylinder and no corners (thus the two sides are opposite). Cylinders still have a *top*, *bottom*, *left side*, *right side* and *center*.

2.1.3 Orb Web Graphs

An *orb web graph* is a cylinder with an extra vertex connected by an edge to all the vertices along one side. The vertex set is thus $\{(x, y) \mid 0 \leq x < h, 0 \leq y < w\} \cup \{(-1, -1)\}$. The edge set is now $\{\{(u, v), (x, y)\} \mid |u - x| + |v - y| = 1\} \cup \{(0, v), (h - 1, v)\} \cup \{(-1, -1), (0, v)\}$. See Figure 2.3.

We define a *row* and *column* and *peripheral vertex* to be the same as they were defined for cylinders. The definition for a *side* has, however, changed to the column where $i = w - 1$. Note that there is only one side on a web graph and no corners. We also define a *hub* to be the vertex labeled $(-1, -1)$. Webs may still have a *top*, *bottom*, *right side* and *center*.

2.1.4 Wall Graphs

Since the search number problem is known to be NP-complete on 3-regular planar graphs, the wall graph, which is essentially a 3-regular cylinder is a very interesting type of graph. Walls are defined only for heights $h \geq 3$ and even widths $w \geq 4$. The

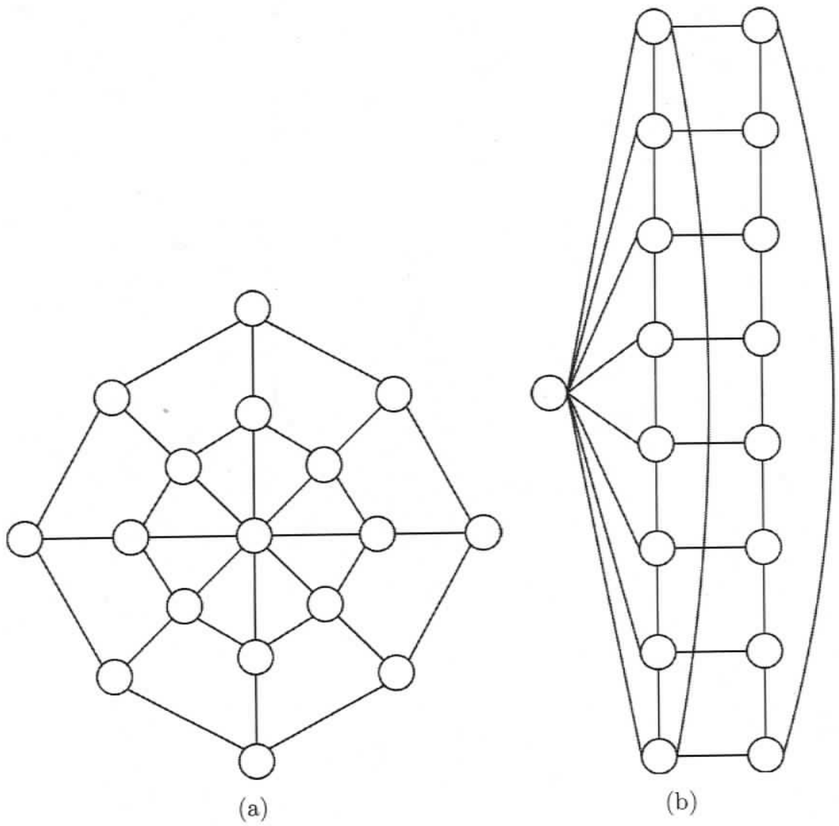


Figure 2.3: An orb web graph, where $h = 8$ and $w = 2$. a) The graph is drawn like a spider's orb web. This layout is where the graph gets its name. b) The graph is drawn like a cylinder. This is the form most commonly used as it is easier to work with.

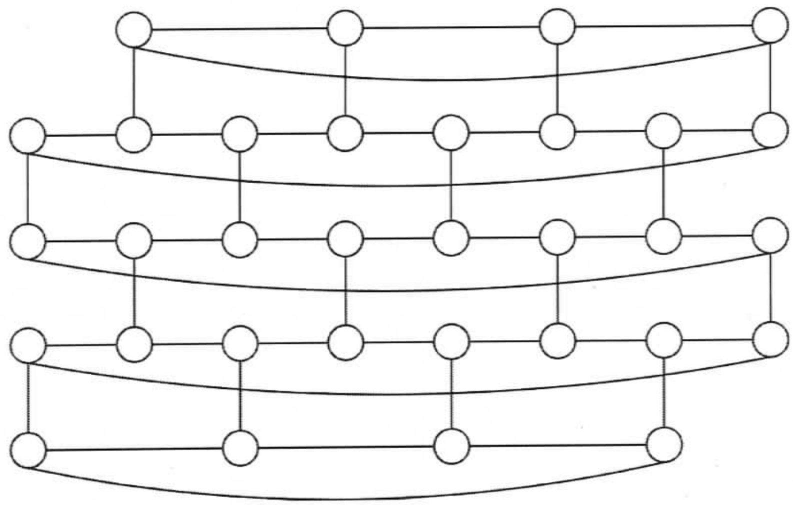


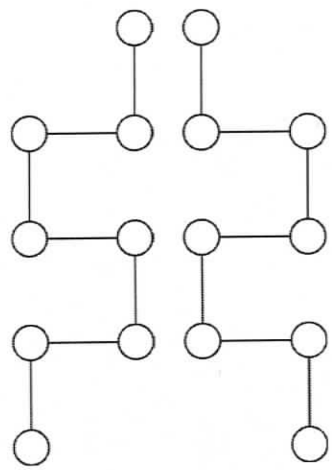
Figure 2.4: A wall graph, where $h = 5$ and $w = 8$

vertex set of a *wall graph* or *wall* is $\{(x, y) \mid 0 \leq x < h, 0 \leq y < w\} \setminus \{\{(0, y) \mid y \text{ is even}\} \cup \{(h-1, y) \mid \text{if } (h-1) \text{ is even then } y \text{ is even else } y \text{ is odd}\}\}$. The edge set is composed of:

- horizontal long row edges, where $1 \leq u \leq h-2$,
  $\{\{(u, v), (u, y)\} \mid v - y = 1\} \cup \{\{(u, 0), (u, w-1)\}\}$
- horizontal top short row edges, $\{\{(0, v), (0, y)\} \mid v - y = 2 \text{ and } y \text{ is odd}\}$
- horizontal bottom short row edges
  $\{\{(h-1, v), (h-1, y)\} \mid v - y = 2 \text{ and } y \text{ is odd if } h \text{ is even and } y \text{ is even if } h \text{ is odd}\} \cup \{\{(h-1, 0), (h-l, w-2)\} \text{ if } h \text{ is odd or } \{(h-1, 1), (h-l, w-1)\} \text{ if } h \text{ is even}\}$
- vertical edges between long rows, where $1 \leq u, x \leq h-2$,
  $\{\{(u, v), (x, v)\} \mid x - u = 1 \text{ and } u \text{ is even and } v \text{ is odd or } u \text{ is odd and } v \text{ is even}\}$
- vertical edges between short rows and long rows

$\{\{(0,v),(1,v)\} \mid v \text{ is odd }\} \cup \{\{(h-2,v),(h-1,v)\} \mid v \text{ is even if } h \text{ is odd and } v \text{ is odd if } h \text{ is even }\}$.

We make a special definition for columns in walls, because there is no continuous sequence of vertical edges. There are two possible sets of columns in any wall. If a column that includes vertex $(0,v)$ also includes vertex $(1,v-1)$ we say that it is *left-going.* If, instead, it includes vertices $(0,v)$ and $(1,v+1)$ we say that it is *right-going.*



(a) (b)

Figure 2.5: A column in a wall. Figure 2.5(a) depicts a left going column and Figure 2.5(b) depicts a right going column.

Like cylinders, walls have two *sides*, namely the short rows. Walls may still have a *top* side and a *bottom* side. Walls do not have a *center*.

The *connections* or *connecting edges* between rows are the vertical edges. Since we picture the row indices as increasing from top to bottom, it will be convenient to refer to *back vertex* or *back connection* for row i, meaning the connections to row $i-1$, and the *fore vertex* or *fore connection*, meaning the connections to row $i+1$.

A *tower* of height i and width k, is a set of clean edges, k per long row such that in every long row, in the range 1 through i, edges $j, j+1, \ldots j+k-1$ are all clean, for some (fixed) j.

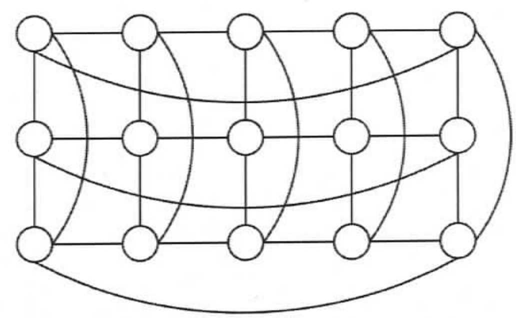


Figure 2.6: A torus graph, where $h = 3$ and $w = 5$

2.1.5 Torus Graphs

The *torus graph* or *torus* (plural is *tori*) is a supergraph of a (h, w)-grid graph with wrap-around edges in both dimensions. The vertex set is $\{\{x, y\} \mid 0 \leq x < h, 0 \leq y < w\}$ and the edge set is $\{\{\{u, v\}, \{x, y\}\} \mid |u - x| + |v - y| = 1\} \cup \{\{0, v\}, \{h - 1, v\}\} \cup \{\{u, 0\}, \{u, w - 1\}\}$. The definitions for a *row* and *column* are the same as for (h, w)-grids but there are now no *sides*, and therefore, no peripheral vertices. Tori still have a *top*, *bottom* and *center*.

2.2 Node Search Number

In the *node search* game, a graph is thought of as a series of *tunnels* through which a fast *fugitive* is moving. Edges in the graph represent the tunnels, while the vertices represent the intersections. The object of the game is to guarantee the capture of the fugitive using the minimum number of *guards*, no matter the movements of the fugitive.

One can guarantee the capture of the fugitive by *cleaning* all the edges of the graph according to the following rules. Initially, all the edges are considered to be *dirty*, meaning that the fugitive could be hiding in any of them. At each step, a new guard is added to a vertex or an existing guard is removed. If ever, for any edge $e =$

$\{x, y\}$, both the vertices x and y contain a guard the edge e is considered to be *cleaned.* The edge e will remain clean unless *recontamination* occurs. Recontamination can occur if one of the guards on x or y, say x, is removed and there exists a dirty edge incident with that vertex. Then all edges incident with x become dirty again. The game ends when all edges are clean.

A sequence of steps that cleans all the edges in a graph is referred to as a *strategy.* At any step t during a strategy S, the *node search number at step t* is the number of guards on the graph at that step. The *node search number* of that strategy on that graph is maximum node search number over all steps. The node search number of a graph, denoted $\text{ns}(G)$ is the minimum node search number over all strategies. A strategy is said to be *progressive* if recontamination does not occur.

It was proven by LaPaugh [LaP93] that, for a very similar related problem called edge searching, recontamination does not help. When Kirousis and Papadimitriou created the node searching problem, they proved, using this result, the following theorem:

Theorem 1 ([KP86]). *For all graphs G, recontamination does not help in node searching.*

Hence, we need to consider only progressive strategies.

It is often convenient to think of a *clean component* which is a subgraph formed by all clean edges and the all vertices which are either incident with a clean edge or guarded.

A row R in a graph $G = (V, E)$ is a *completely clean row* if there does not exist an edge $e \in ((R \times R) \cap E)$ that is dirty. Similarly, for a *completely clean column C* in G, no edge $e \in ((C \times C) \cap E)$ is dirty. A *partly clean row* or a *partly dirty row* is a row that contains both clean and dirty edges. Similarly, a *partly clean column* or a *partly dirty column* is a column that contains both clean and dirty edges. A

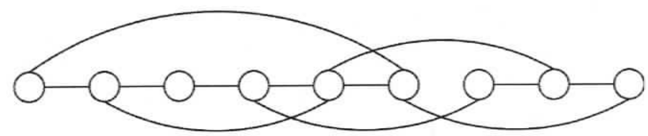


Figure 2.7: A linear layout. The vertex separation of the layout is 3

completely dirty row is a row that contains no clean edges, while a *completely dirty column* is a column that contains no clean edges. An *almost clean row* is a row with one or two dirty edges. An *almost clean column* is a column with one or two dirty edges. Any vertex adjacent to both a clean and a dirty edge must guarded. We call such a vertex a *critical vertex*. We sometimes refer to an *exterior vertex* which is a critical vertex except that the vertex is incident with both a clean edge and a dirty edge in the same row. An *exterior guard* is a guard on an exterior vertex.

2.3 Vertex Separation

A *vertex separator* or *separator* of a connected graph is a set of vertices whose removal separates the graph into two or more components. For a graph G, a *linear layout* or *layout* is a bijective mapping $L : V(G) \rightarrow \{1, 2, \ldots, |V(G)|\}$. A *partial layout* of G is a bijective mapping $L : V' \rightarrow \{1, 2, \ldots, |V'|\}$ where $V' \subseteq V(G)$. For a partial layout L, let $V_L(i) = \{x \mid x \in V$ and there exists $y \in V$ such that $\{x, y\} \in E$ and $L(x) \leq i$ and either $L(y) > i$ or $L(y)$ is undefined$\}$. The *vertex separation* of G with respect to L, denoted by $\text{vs}_L(G)$, is defined by $\text{vs}(G, L) = \max\{|V_L(i)| 1 \leq i \leq |\text{domain}(L)|\}$ and the vertex separation of G is defined by $\text{vs}(G) = \min\{\text{vs}_L(G) \mid L$ is a layout of $G\}$.

Node searching is equivalent to vertex separation, thus providing a useful alternative for examining properties of the problem. Kirousis and Papadimitriou give the following theorem.

Theorem 2 ([KP86]). $\text{ns}(G) = \text{vs}(G) + 1$

The order, from left to right, of the vertices in the layout correspond to the order that guards are placed on the graph in a strategy, and *vice versa*.

2.4 Pathwidth

Given a graph $G = (V, E)$ and a sequence $X_1, X_2, \ldots, X_n \subseteq V$ is a *path-decomposition* of G if the following conditions are satisfied:

1. $\cup X_i = V$
2. for every edge $e = (x, y)$, there exists X_i such that $x, y \in X_i$
3. for $i \leq j \leq k \leq n$, $X_i \cap X_k \subseteq X_j$

The *pathwidth* of G with respect to a path-decomposition is $\text{pw}(G, X_1, X_2, \ldots, X_n) = \max_{i \in [n]} |X_i| - 1$. The pathwidth of a graph, denoted $\text{pw}(G)$ is the minimum of the pathwidth over all path-decompositions.

Pathwidth is a particularly important concept, so it is worth noting that it is equivalent to vertex separation and hence to node search number. Kinnersley showed that $\text{vs}(G) = \text{pw}(G)$ and that layouts can be derived from path-decompositions and *vice versa* [Kin92]. Pathwidth is an important concept in the theory of parameterized complexity. It is possible that an NP-complete problem can be consider to run in polynomial time if the pathwidth is considered to be fixed. The implication is that the problem may then be tractable if the pathwidth of a particular instance is small.

Chapter 3

Results for Node Search Number

In this chapter we give matching upper and lower bounds for the node search number on the five types of grid-like graphs. The upper bounds are intuitively obvious but the proofs of the lower bounds are not trivial. Further, we found that the obvious upper bound on certain tori is not in fact optimal.

We need only consider optimal strategies that are progressive, see Theorem 1, so a clean component in the graph will never decrease in size. We use this fact to prove the lower bounds, observing that once a component has grown to a certain size, usually large enough to contain one or more rows and columns, that some minimum number of guards are needed to prevent recontamination.

3.1 Node Search Number on (h, w)-Grid Graphs

Most of the graphs that we examine are variations of the basic (h, w)-grid. We examine this type of graph first. The proof is in two parts, first a lemma that describes an obvious search strategy, giving an upper bound. This is followed by another lemma that gives a matching lower bound.

Lemma 1. *On any (h, w)-grid G,* $\mathrm{ns}(G) \leq \min(h+1, w+1)$.

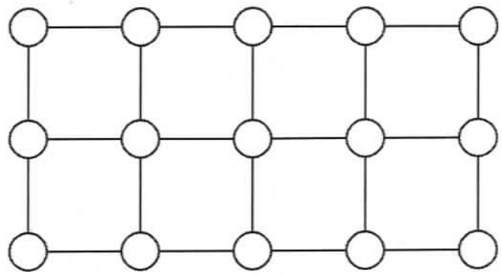


Figure 3.1: A (h,w)-grid graph, where $h = 3$ and $w = 5$.

Proof. Consider the following search strategy: place a guard on every vertex in the leftmost column. This requires h guards.

Place another guard x in the second column from the left, we now have used $h + 1$ guards. Remove the guard y that is on the vertex adjacent to x in the first column. Continue in this fashion until the second column from the left is filled with guards, the first column on the left will now have no guards. This step will require exactly $h + 1$ guards and in the end we will have exactly h guards.

We can continue like this, moving from one column to the adjacent completely dirty column until the whole grid is clean. We use $h+1$ guards. For the width, perform the same strategy except substituting width for height and rows for columns. The upper bound will be the smaller of these results, thus, $\text{ns}(G) \leq \min(h+1, w+1)$. □

Lemma 2. *For any (h, w)-grid G,* $\text{ns}(G) \geq \min(h + 1, w + 1)$.

Proof. Let step t be the first step at which the set of clean edges defines a component C^t, which includes one column or one row.

Suppose C^t includes a column. At the completion of step t, all rows must contain at least one vertex in C^t, because there is a column in C^t. Therefore at step $t - 1$, at least $h - 1$ rows contain at least one vertex in C^{t-1}. Step t is a placement because it increases the number of clean edges.

C^{t-1} cannot contain a row, otherwise a row would be clean prior to step t. Since

C^{t-1} does not contain a row, every row must contain a dirty edge. Since at least $h-1$ rows contain a vertex in C^{t-1}, at least $h-1$ rows contain a vertex in C^{t-1} which is incident with a dirty edge. Each such vertex must be occupied by a guard. Hence, there are at least $h-1$ guards on $h-1$ rows and adjacent to both a clean and a dirty edge at the completion of step $t-1$.

Hence, if there are more than $h-1$ guards on C^{t-1}, we are done.

Suppose there are exactly $h-1$ guards on C^{t-1}. Then one row, say R, contains no guards. No row is completely clean before step t. Hence, all the edges in R are dirty. Step t cleans at least one edge in a column, say column H, so that all edges in H are in C^t. Suppose that last placement is to vertex v. The vertex v must be in R, because, if not, there remains a vertex in H with no guard and adjacent to a dirty row edge. Then some edge in H must still be dirty.

Consider any row edge incident with v. Such an edge is in R and hence is not in C^{t-1}. Nor does the last placement clean any edge in R, because R contains no guards. Hence, v is adjacent to a dirty edge. Since R is completely dirty the adjacent rows, since they only have one guard each, must also be completely dirty. Hence, the guards in H that are adjacent to v remain critical. Hence, there are h guards on C^t, all critical. Hence, the next move cannot be a removal, so $h+1$ guards are on C^{t+1}.

By identical reasoning, if C^t first includes a row then there are at least $w+1$ guards on the graph at the completion of step t or $t+1$. Since one or the other case must obtain, the lemma follows. □

Theorem 3. *On any (h, w)-grid graph G,* $\text{ns}(G) = \min(h+1, w+1)$.

Proof. From Lemma 1 we get an upper bound $\text{ns}(G) \leq \min(h+1, w+1)$ and from Lemma 2 we get a lower bound $\text{ns}(G) \geq \min(h+1, w+1)$, therefore, $\text{ns}(G) = \min(h+1, w+1)$. □

3.2 Node Search Number on Cylinders

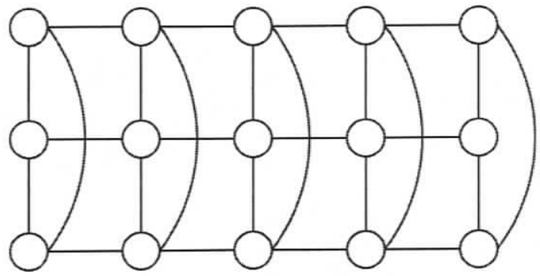


Figure 3.2: A cylinder graph, where $h = 3$ and $w = 5$. Note that the wrap-around edges are in the columns.

Lemma 3. *If G is a cylinder then* $\text{ns}(G) \leq \min(h+1, 2w+1)$.

Proof. Consider the following search strategy: place a guard on every vertex in the left-most column. This requires h guards.

Place another guard x in the second column from the left, we now have used $h+1$ guards. Remove the guard y that is on a vertex in the first column that is adjacent to x. Continue in this fashion until the second column from the left is filled with guards, the first column on the left will now have no guards. This step will require exactly $h+1$ guards and in the end we will have exactly h guards.

We can continue like this, moving from one column to the adjacent completely dirty column until the whole grid is clean. We use $h+1$ guards.

Alternatively, consider the following search strategy: place a guard on every vertex in the first and second row from the top. This requires $2w$ guards. Place another guard x in the third row from the top, we now have used $2w+1$ guards. Remove the guard y that is on a vertex in the third row that is adjacent to x. Continue in this fashion until the third row from the top is filled with guards, the second row from the top will now have no guards, but the first column from the top will still have guards. This step will require no more than $2w + 1$ guards and in the end we will have no more than $2w$ guards.

We can continue like this, moving from one row to the adjacent completely dirty row until the whole grid is clean. We will never use more than $2w + 1$ guards.

Depending on the dimensions of the graph one of these search strategies may be superior or they may be equivalent. Thus, $\text{ns}(G) \leq \min(h + 1, 2w + 1)$. □

Lemma 4. *Let G be a cylinder,* $\text{ns}(G) \geq \min(h + 1, 2w + 1)$.

Proof. Let step t be the first step at which the set of clean edges defines a component C^t which includes one column (including wrap-around edges) or two rows.

- Suppose C^t contains a column, say H. At the completion of step $t - 1$ there is no completely clean column and there is no more than one completely clean row.

  - Consider the case where at step $t - 1$ one row is clean, say R. Therefore, there must be $h - 1$ rows that contain a dirty edge.

    All rows must contain at least one vertex in C^t, because there is a completely clean column. Therefore, at least $h - 1$ rows contain at least one vertex in C^{t-1}.

    Hence, there must be at least $h-2$ verticies in C^{t-1} that are incident with both a clean and a dirty row edge. Each such vertex must be occupied by a guard. Hence, there are at least $h - 2$ guards on $h - 2$ rows at the completion of step $t - 1$.

    R is the only completely clean row at step $t - 1$, hence it is adjacent to a row containing at least one dirty edge incident with an unguarded vertex. Hence, R must contain at least one guard adjacent to that unguarded vertex. Hence, at the completion of step $t-1$ there must be at least $h-1$ guards, all critical, on $h-1$ rows.

- Consider the case where at step $t-1$ where there are no completely clean rows. Therefore, there exist h rows that contain a dirty edge. All rows must contain at least one vertex in C^t, because there is a completely clean column. Therefore, at least $h-1$ rows contain at least one vertex in C^{t-1}. Hence, there must be at least $h-1$ verticies in C^{t-1} that are incident with both a clean and a dirty edge. Each such vertex must be occupied by a guard. Hence, there are at least $h-1$ guards on $h-1$ rows at the completion of step $t-1$, all critical.

So, regardless of the presence of a completely clean row, there must be at least $h-1$ critical guards on $h-1$ rows at the completion of step $t-1$. Step t is a placement because it increases the number of clean edges. Hence, if there are more than $h-1$ guards on C^{t-1}, we are done.

Suppose there are exactly $h-1$ critical guards on C^{t-1}. Then one row, say R, contains no guards. No row is completely clean before step t. Hence, all the edges in R are dirty. Step t cleans at least one edge in a column, say column H, so that all edges in H are in C^t. Suppose that last placement is to vertex v. The vertex v must be in R, because, if not, there remains a vertex in H with no guard and adjacent to a dirty row edge. Then some edge in H must still be dirty.

Consider any row edge incident with v. Such an edge is in R and hence is not in C^{t-1}. Nor does the last placement clean any edge in R, because R contains no guards. Hence, v is adjacent to a dirty edge. Since R is completely dirty the adjacent rows, since they only have one guard each, must also be completely dirty. Hence, the guards in H that are adjacent to v remain critical. Hence, there are h guards on C^t, all critical. Hence, the next move cannot be a removal, so $h+1$ guards are on C^{t+1}.

- Suppose at step t two rows are clean and no column is clean. Then all columns must contain a dirty edge.
  - Consider every column that contains a clean edge and a dirty edge. Because the columns have wrap-around edges, there must be at least two critical vertices in any such column.
  - Consider every column that contains only dirty edges. Every such column contains two vertices that are incident with both a clean edge (in the rows) and a dirty edge (in the column). These vertices are critical and hence must be occupied by a guard.

Thus, there are at least two critical guards per column. So, at the completion of step t, there must be at least $2w$ guards on the graph. If there are more than $2w$ guards we are done. If there are exactly $2w$ guards then the next step is a placement because they are all critical. □

Theorem 4. *If G is a cylinder then* $\mathrm{ns}(G) = \min(h+1, 2w+1)$.

Proof. From Lemma 3 we get an upper bound $\mathrm{ns}(G) \leq \min(h+1, 2w+1)$ and from Lemma 4 we get a lower bound $\mathrm{ns}(G) \geq \min(h+1, 2w+1)$, therefore, $\mathrm{ns}(G) = \min(h+1, 2w+1)$. □

3.3 Node Search Number on Orb Webs

The orb webs are similar to cylinders. The only difference is the addition of the hub vertex. However, the hub vertex does have an effect on the node search number.

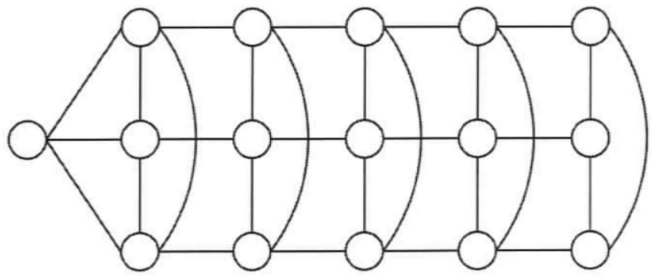


Figure 3.3: An orb web, where $h = 3$ and $w = 5$.

Lemma 5. *Let G be an orb web,* $\text{ns}(G) \leq \min(h+1, 2w+2)$.

Proof. Consider the following search strategy, see Figure 2.3 b). Place a guard on the hub vertex. Then place a guard on every vertex in the left-most column. This requires $h + 1$ guards. We can now remove the guard on the hub vertex without recontamination occurring. There are now h guards on the graph.

Place another guard x in the second column from the left, we now have used $h+1$ guards. Remove the guard y that is on a vertex adjacent to x. Continue in this fashion until the second column from the left is filled with guards, the first column on the left will now have no guards. This step will require no more than $h+1$ guards and at the end we will have no more than h guards.

We can continue like this, moving from one column to the adjacent dirty column until the whole grid is clean. We will never use more than $h + 1$ guards.

Alternatively, consider the following search strategy. Place a guard on every vertex in the first and second row from the top as well as the hub vertex. This requires $2w + 1$ guards. Place another guard x in the third row from the top, we now have used $2w + 2$ guards. Remove the guard y that is on a vertex adjacent to x. Continue in this fashion until the third row from the top is filled with guards, the second row from the top will now have no guards, but the first column from the top will still have guards. A guard must also remain on the hub vertex, otherwise recontamination will occur. This step will require no more than $2w + 2$ guards and at the end we will have no more than $2w + 1$ guards.

We can continue like this, moving from one row to the adjacent dirty row until the whole grid is clean. We will never use more than $2w + 2$ guards. Depending on the dimensions of the graph one of these search strategies may be better or they may be equivalent. Thus, $\text{ns}(G) \leq \min(h + 1, 2w + 2)$. □

Lemma 6. *Let G be an orb web,* $\text{ns}(G) \geq \min(h + 1, 2w + 2)$.

Proof. Let step t be the first step at which the set of clean edges defines a component C^t which includes one column (including wrap-around edge) or two rows and the hub.

- Suppose C^t contains a column, say H, at step t.

  The situation at the completion of step $t - 1$ must be the complement of the situation at t. Thus either C^{t-1} contains neither a column nor two rows, or C^{t-1} contains neither a column nor the hub.

  - Consider the case where C^{t-1} contains no column and less than two rows. The argument for this case is identical to the first part of the argument in the proof of Lemma 4 for cylinders. Therefore, there are at least $h - 1$ guards on at least $h - 1$ rows.

  - Consider the case where C^{t-1} contains no columns, and no hub.

    Since the hub is not in C^{t-1}, all edges incident to it must be dirty. Since every edge incident with the hub is incident to a row, all rows must contain a guard or be adjacent to a dirty edge. So for every completely clean row, there must be a guard.

    Every dirty row contains a dirty edge. If the dirty row contain a vertex in C^{t-1}, then there exists a vertex incident with both a clean and a dirty edge. Hence, every dirty row with a vertex in C^{t-1} must contain a guard.

Since there is a column in C^t, there must be at least $h-1$ rows that contain a vertex in C^{t-1}. Every clean row must contain a vertex in C^{t-1}, and every clean row contains a guard. The remaining rows must be dirty, but if they also contain a vertex in C^{t-1} they must have a guard. So, there must be at least $h-1$ guards on $h-1$ rows at step $t-1$.

So, regardless of the presence of a clean row, there must be $h-1$ guards on $h-1$ rows at the completion of step $t-1$. By the same reasoning as the second part of the proof of Lemma 2 we can conclude that there must be at least $h+1$ guards on the graph at t or $t+1$.

- Suppose at step t two rows and the hub are in C^t, but no column is completely clean. Hence, every column must contain a dirty edge.
  - Consider every column that contains a clean edge and a dirty edge. Because the columns have wrap-around edges, there must be at least two critical guards in the column.
  - Consider every column that contains only dirty edges. Every such column contains two vertices that are incident to both a clean edge (in the rows) and a dirty edge (in the column). Each such vertex must be occupied by a guard.

Thus, there are at least two critical guards per column. So, at the completion of step t, there must be at least $2w$ critical guards on the graph. The hub is adjacent to an entire column. Since all columns are dirty, the column adjacent to the hub must contain a dirty edge. Since it contains a dirty edge, there must be an unguarded vertex in that column and the hub is adjacent to that vertex. Therefore, since the hub is in C^t, the hub must be guarded. Hence, there must be at least $2w+1$ critical guards on the graph at the completion of step t.

Of these guards, none can be removed before the next placement step because they are all critical. Hence, at the next placement step there must be at least $2w + 2$ guards on the graph. □

Theorem 5. *If a graph G is a web graph then* $\mathrm{ns}(G) = \min(h + 1, 2w + 2)$.

Proof. From Lemma 5 we get an upper bound $\mathrm{ns}(G) \leq \min(h + 1, 2w + 1)$ and from Lemma 6 we get a lower bound $\mathrm{ns}(G) \geq \min(h + 1, 2w + 1)$. Therefore, $\mathrm{ns}(G) = \min(h + 1, 2w + 1)$. □

3.4 Node Search Number on Walls

The strategy to clean walls, is more complicated, requiring some very careful choices when placing and removing guards. One of the major complications when trying to prove walls is that there are two types of rows, long and short. The existence of short rows make the proof for the lower bounds much more complicated then if it had just had long rows.

3.4.1 Wall Upper Bounds

Lemma 7. *Let G be a wall,* $\mathrm{ns}(G) \leq \min(2h + 1, w/2 + 2)$.

Proof. Consider the following search strategy: place a guard on every vertex in the first column from the left. This requires $2h - 2$ guards.

We continue, we need to define a *rear guard vertex* to be a vertex u that is incident with exactly two clean edges. We observe, that all such vertices contain a guard, since every vertex is degree 3, and we will refer that guard as the *rear guard.* We also define a *vanguard vertex* to be a vertex v that is adjacent to a rear guard vertex u such that the edge $\{v, u\}$ is dirty.

Place guards in the second column from the left. If any vertex in this column is a vanguard vertex then place a guard there first, and remove the rear guard immediately. The only non-vanguard vertices in the column are the vertices in the short rows. The vanguard vertices, regardless of how many are present, never increase the number of guards by more than one because they are always followed by a removal. As a result, the number guards should not have exceeded $2h - 2 + 2 + 1 = 2h + 1$, and $2h$ guards should remain on the graph after the third and fourth columns are clean.

We can continue like this, moving from one column to the adjacent dirty column until the whole graph is clean. All vertices from this point forward are vanguard vertices, thus, no more than $2h + 1$ guards are needed to clean the remainder of the graph.

Alternatively, consider the following search strategy: place a guard on every vertex in the top row. This requires $w/2$ guards (one per column).

Place another guard x on a vanguard vertex in the second row from the top. Remove the rear guard. Continue in this fashion until the second row from the top has no more vanguard vertices. A guards should now be placed on any one of the unguarded vertices in this row. One of the vertices adjacent to this vertex on the same row will become a vanguard vertex. Place a guard on the vanguard vertex, at this point there should be $w/2 + 2$ vertices on the graph. Remove the rear guard. There will be another vanguard on this row (if there are any dirty edges remaining) which we can place a guard on and remove the rear guard, which causes another vanguard to appear, etc.... The remainder of the edges can be cleaned by placing a guard on the vanguard vertex and removing the rear guard. This whole row can be cleaned for $w/2 + 2$ vertices, and $w/2$ vertices will remain (one per column).

We can continue like this, moving from one row to the adjacent dirty row until the whole graph is clean. Guards should always be placed on the vanguard vertices first, followed by the immediate removal of the rear guard. This will require $w/2+2$ guards.

Depending on the dimensions of the graph one of these search strategies may be better or they may be equivalent. Thus, $\text{ns}(G) \leq \min(2h+1, w/2+2)$. □

3.4.2 Completely Dirty Short Row Lemmas

Lemma 8. *In a wall, if one short row is completely dirty and contains exactly one guard, and all long rows contain exactly two guards, then long row i can contain no more than $2i+2$ clean edges where a row is numbered i if there are $i-1$ rows between it and the completely dirty short row.*

Proof. Extended lemma:

1. if the two guards contained in long row i are on back vertices, then long row i contains no more than $2i+2$ clean edges.

2. if the two guards contained in long row i are on fore vertices, then long row i contains no more than $2i$ clean edges.

3. if one guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i+1$ clean edges.

In all the above cases, the number of clean edges is less than or equal to $2i+2$, thus, the extended lemma implies the lemma.

We will prove that any long row i cannot have more than $2i+2$ clean edges by induction on i.

Base Case. $i = 1$

From Figure 3.4 it is easy to see that the extended lemma holds for all three cases.

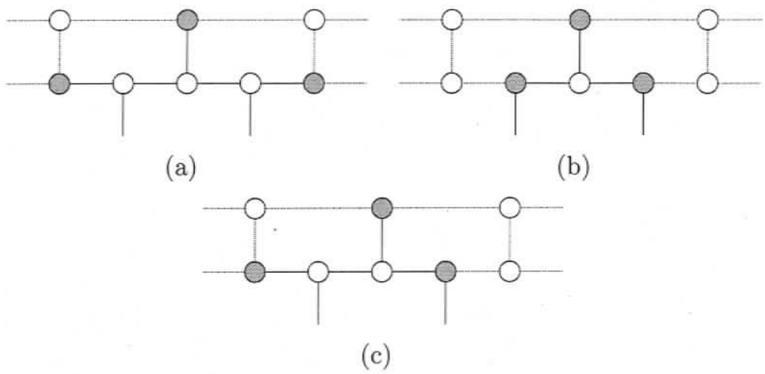


Figure 3.4: **The extended lemma holds for 3.4(a) both guards are on back vertices, 3.4(b) both guards on fore vertices and 3.4(c) one guard on a back vertex and one guard on a fore vertex.**

Induction. Let us assume that the extended lemma is true for all i, $1 \leq i \leq i_0$, for some $i_0 \geq 1$. We show that it is then true for $i_0 + 1$, and hence true for all i.

Case 1: *Suppose both guards contained in long row* i_0 *are on back vertices*

By the inductive hypothesis, long row i_0 contains no more than $2i_0 + 2$ clean edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

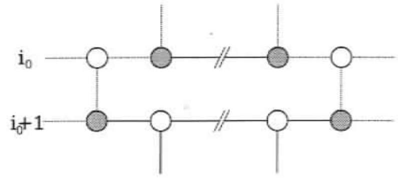
Case 1.1 *Both guards contained in long row* $i_0 + 1$ *are on back vertices*

From Figure 3.5(a) we see that no more than 2 clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ clean edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

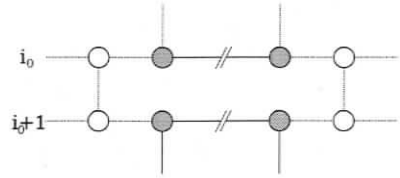
Case 1.2 *Both guards contained in long row* $i_0 + 1$ *are on fore vertices*

From Figure 3.5(b) we see that no clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ clean edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.

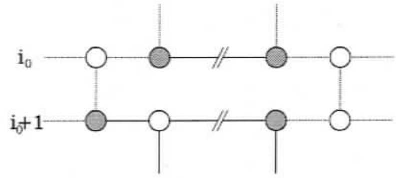
Case 1.3 *One guard contained in long row* $i_0 + 1$ *is on a back vertex and the other guard is on a fore vertex*



(a) Long row $i_0 + 1$ can contain at most two more clean edges than long row i_0.



(b) Long row i_0+1 can contains the same number of clean edges as long row i_0.



(c) Long row $i_0 + 1$ can contain at most one more clean edge than long row i_0.

Figure 3.5: The various configuration of guards in Case 1

From Figure 3.5(c) we see that no more than 1 clean edge can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ clean edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma.

Case 2: *Suppose both guards contained in long row* i_0 *are on fore vertices*

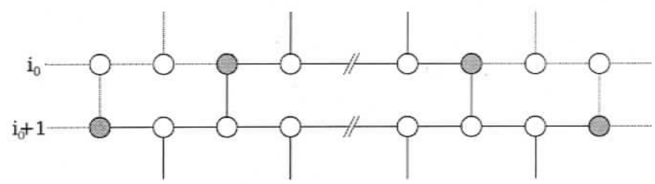
By the inductive hypothesis, long row i_0 contains no more than $2i_0$ clean edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

Case 2.1 *Both guards contained in long row* $i_0 + 1$ *are on back vertices*

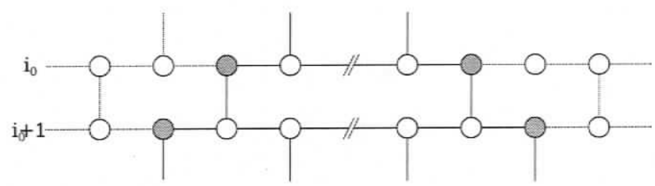
From Figure 3.6(a) we see that no more than 4 clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ clean edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

Case 2.2 *Both guards contained in long row* $i_0 + 1$ *are on fore vertices*

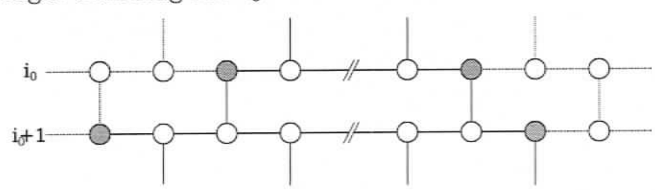
From Figure 3.6(b) we see that no more than 2 clean edges can be gained.



(a) Long row $i_0 + 1$ can contain at most four more clean edges than long row i_0.



(b) Long row $i_0 + 1$ can contains at most two more clean edges than long row i_0.



(c) Long row $i_0 + 1$ can contain at most three more clean edges than long row i_0.

Figure 3.6: The various configuration of guards in Case 2

Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ clean edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.

Case 2.3 *One guard contained in long row $i_0 + 1$ is on a back vertex and the other guard is on a fore vertex*

From Figure 3.6(c) we see that no more than 3 clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ clean edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma.

Case 3: *Suppose one guard contained in long row i_0 is on a back vertex and one guard contained in long row i_0 is on a fore vertex*

By the inductive hypothesis, long row i_0 contains $2i_0 + 1$ clean edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

Case 3.1 *Both guards contained in long row $i_0 + 1$ are on back vertices*

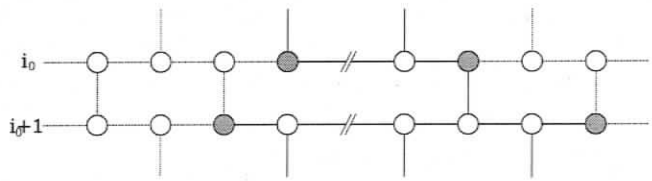
From Figure 3.7(a) we see that no more than 3 clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ clean edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

Case 3.2 *Both guards contained in long row $i_0 + 1$ are on fore vertices*

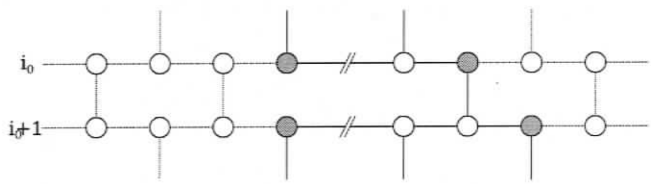
From Figure 3.7(b) we see that no more than 1 clean edge can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ clean edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.

Case 3.3 *One guard contained in long row $i_0 + 1$ is on a back vertex and the other guard is on a fore vertex*

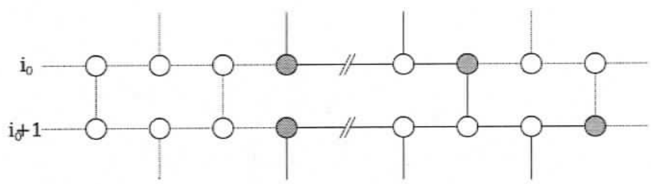
From Figure 3.7(c) and Figure 3.7(d) we see that no more than 2 clean edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ clean edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma. □



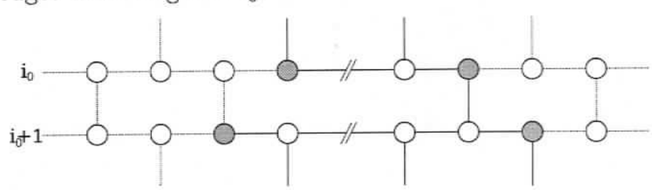
(a) Long row $i_0 + 1$ can contain at most three more clean edges than long row i_0.



(b) Long row $i_0 + 1$ can contains at most one more clean edge than long row i_0.



(c) Long row $i_0 + 1$ can contain at most two more clean edges than long row i_0.



(d) Long row $i_0 + 1$ can contain at most two more clean edges than long row i_0.

Figure 3.7: The various configuration of guards in Case 3

Corollary 1. *In a wall, if one short row is completely dirty and contains exactly one guard, and all long rows contain exactly two guards, then no long row can contain more than $2h - 2$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 8 provides an upper bound of $2h - 2$ as the most clean edges that any long row can contain. □

Lemma 9. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly three guards, and all other long rows contain exactly two guards, then long row i can contain no more than $2i+4$ clean edges where a row is numbered i if there are $i - 1$ rows between it and the completely dirty short row.*

Proof. Extended lemma:

1. if the two exterior guards contained in long row i are on back vertices, then long row i contains no more than $2i + 4$ clean edges.

2. if the two exterior guards contained in long row i are on fore vertices, then long row i contains no more than $2i + 2$ clean edges.

3. if one exterior guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i + 3$ clean edges.

In all the above cases, the number of clean edges is less than or equal to $2i + 4$, thus, the extended lemma implies the lemma.

We will prove that any long row i cannot have more than $2i + 4$ clean edges by induction on i.

Let long row k be the one that contains three guards.

Base Case. There are two possible cases for the base case. Either $k = 1$, meaning the long row with three guards is adjacent to the completely dirty short row or $k > 1$

meaning that the long row with three guards is adjacent only to long rows containing two guards.

Case $i = k = 1$

From Figure 3.8 it is easy to see that the extended lemma holds for all three cases.

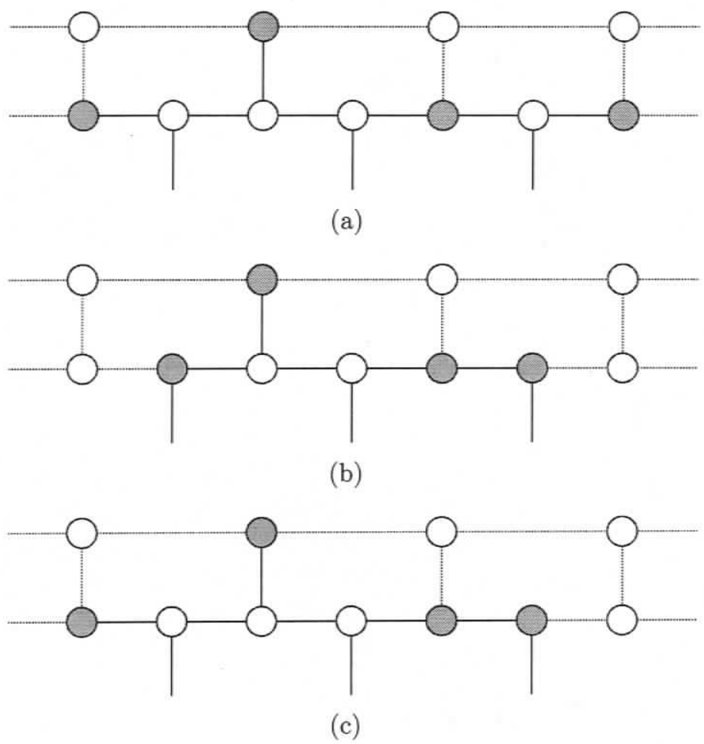


Figure 3.8: Because $i = k = 1$ the extended lemma holds for 3.8(a) both exterior guards are on back vertices, 3.8(b) both exterior guards on fore vertices and 3.8(c) one exterior guard is on a back vertex and one exterior guard is on a fore vertex.

Case $i = k > 1$

The guards in long row $i - 1$ can be either on back vertices, fore vertices and on both a back and a fore vertex.

Consider the following three cases:

Case 1. *Both guards on long row* $i-1$ *are on back vertices*

From Lemma 8's extended lemma item 1 we know that there cannot be more than $2(i-1)+2 = 2i$ clean edges on long row $i-1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.

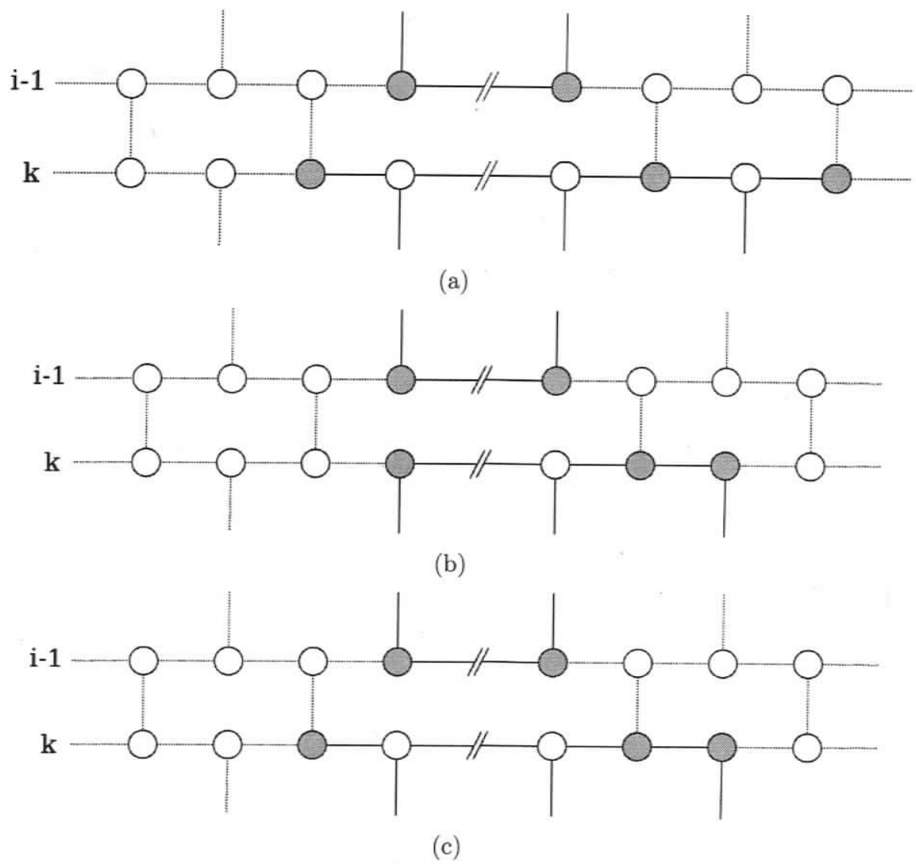


Figure 3.9: The various configuration of guards in Case 1

Case 1.1 *Both exterior guards on long row* k *are on back vertices.*

From Figure 3.9(a) we see that no more than 4 clean edges can be gained. Hence, long row i contains no more than $2i + 4$ clean edges, conforming to item 1 in the extended lemma.

Case 1.2 *Both exterior guards guards on long row k are on fore vertices.*

From Figure 3.9(b) we see that no more than 2 clean edges can be gained. Hence, long row i contains no more than $2i + 2$ clean edges, conforming to item 2 in the extended lemma.

Case 1.3 *One exterior guard on long row k is on a back vertex, the other is on a fore vertex.*

From Figure 3.9(c) we see that no more than 3 clean edges can be gained. Hence, long row i contains no more than $2i + 3$ clean edges, conforming to item 3 in the extended lemma.

Case 2. *Both guards on long row $i - 1$ are on fore vertices*

From Lemma 8's extended lemma item 2 we know that there cannot be more than $2(i-1) = 2i-2$ clean edges on long row $i-1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.

Case 2.1 *Both exterior guards on long row k are on back vertices.*

From Figure 3.10(a) we see that no more than 6 clean edges can be gained. Hence, long row i contains no more than $2i + 4$ clean edges, conforming to item 1 in the extended lemma.

Case 2.2 *Both exterior guards guards on long row k are on fore vertices.*

From Figure 3.10(b) we see that no more than 4 clean edges can be gained. Hence, long row i contains no more than $2i + 2$ clean edges, conforming to item 2 in the extended lemma.

Case 2.3 *One exterior guard on long row k is on a back vertex, the other is on a fore vertex.*

From Figure 3.10(c) we see that no more than 5 clean edges can be

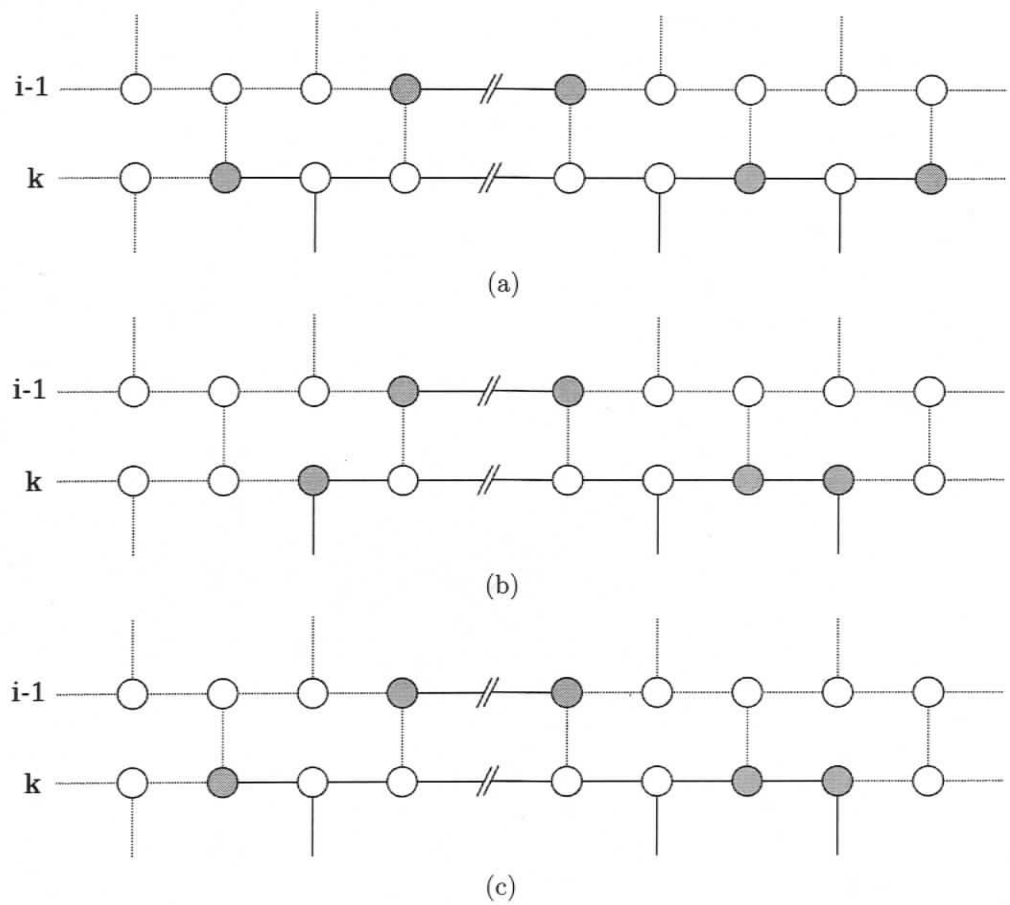


Figure 3.10: The various configuration of guards in Case 2

gained. Hence, long row i contains no more than $2i+3$ clean edges, conforming to item 3 in the extended lemma.

Case 3. *One guard on long row $i-1$ is on a back vertex, the other is on a fore vertex*

From Lemma 8's extended lemma item 3 we know that there cannot be more than $2(i-1)+1 = 2i-1$ clean edges on long row $i-1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.

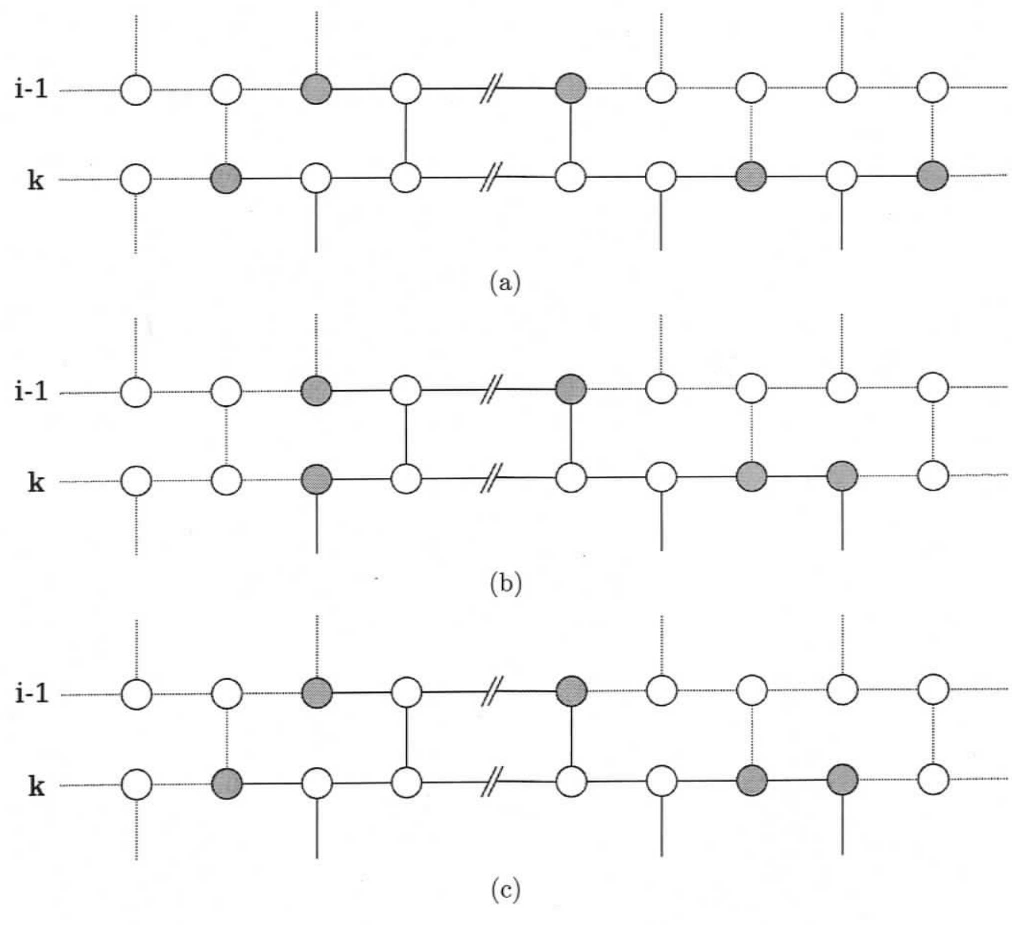


Figure 3.11: The various configuration of guards in Case 3

Case 3.1 *Both exterior guards on long row k are on back vertices.*

From Figure 3.11(a) we see that no more than 5 clean edges can be gained. Hence, long row i contains no more than $2i + 4$ clean edges, conforming to item 1 in the extended lemma.

Case 3.2 *Both exterior guards guards on long row k are on fore vertices.*

From Figure 3.11(b) we see that no more than 3 clean edges can be gained. Hence, long row i contains no more than $2i + 2$ clean edges, conforming to item 2 in the extended lemma.

Case 3.3 *One exterior guard on long row k is on a back vertex, the other is on a fore vertex.*

From Figure 3.11(c) we see that no more than 4 clean edges can be gained. Hence, long row i contains no more than $2i + 3$ clean edges, conforming to item 3 in the extended lemma.

Induction. After long row k all remaining long rows have 2 guards and thus the inductive step is essentially the same as the inductive step of Lemma 8. □

Corollary 2. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly three guards, and all other long rows contain exactly two guards, then no long row can contain more than $2h$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 9 provides an upper bound of $2h$ as the most clean edges that any long row may contain. □

Lemma 10. *In a wall, if one short row is completely dirty and contains exactly one guard, and two long rows contain exactly three guards, and all other long rows contain exactly two guards, then long row i can contain no more than $2i+6$ clean edges where a row is numbered i if there are $i - 1$ rows between it and the completely dirty short row.*

Proof. This proof follows that same pattern as Lemma 9 except let k be the second long row that contains three guards and instead of citing Lemma 8 one needs to cite Lemma 9. □

Corollary 3. *In a wall, if one short row is completely dirty and contains exactly one guard, and two long rows contain exactly three guards, and all other long rows contain exactly two guards, then no long row can contain more than $2h + 2$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 10 provides an upper bound of $2h + 2$ as the most clean edges that any long row may contain. □

Lemma 11. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then long row i can contain no more than $2i+6$ clean edges where a row is numbered i if there are $i - 1$ rows between it and the completely dirty short row.*

Proof. This proof follows that same pattern as Lemma 9 except instead of a long row with three guards there is a long row with four guards. □

Corollary 4. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than $2h + 2$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 11 provides an upper bound of $2h + 2$ as the most clean edges that any long row may contain. □

Lemma 12. *In a wall, a long row with two guards that is adjacent to a completely dirty short row with more than one guards cannot contain more clean edges than a long row with two guards adjacent to a short row with one guard.*

Proof. If any guards on the completely dirty row are adjacent then the completely dirty row has a clean edge, a contradiction. Thus, all the guards on the short row must be at least two edges apart.

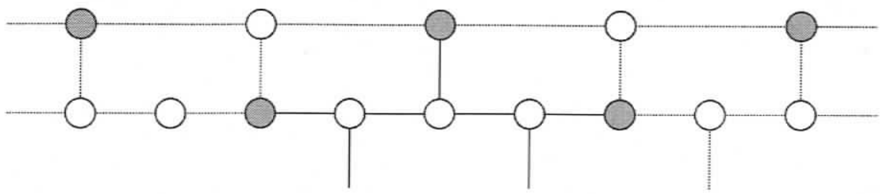


Figure 3.12: The extra guards on the short row does not help protect the adjacent row.

Since every short row edge is equivalent to two long row edges the guards in the long row cannot take advantage of more than one short row guard for the purpose of cleaning edges, see Figure 3.12.

Hence, more than one guard in the short row is no more helpful than one guard. □

Lemma 13. *In a wall, if one short row is completely dirty and contains exactly two guards, and all long rows contain exactly two guards, then long row i can contain no more than $2i + 2$ clean edges where i is the number of long rows plus one from the completely dirty short row.*

Proof. If the completely dirty short had just one guard we know from Lemma 8 the adjacent long row could have at most 4 clean edges. By Lemma 12 we know that even with two guards on the short row, the adjacent long row still has no more than 4 clean edges. Hence, the extra guard in the short row does not effect the number of clean edges in the other rows. Thus, by Lemma 8, long row i can contain at most $2i + 2$ clean edges. □

Corollary 5. *In a wall, if one short row is completely dirty and contains exactly two* *guards, and all long rows contain exactly two guards, then no long row can contain more than $2h-2$ clean edges.*

Proof. There are $h-2$ long rows. Setting i to $h-2$ in Lemma 13 provides an upper bound of $2h-2$ as the most clean edges that any long row may contain. □

Lemma 14. *In a wall, if one short row is completely dirty and contains exactly two guards, one long row contain exactly three guards and all long rows contain exactly two guards, then long row i can contain no more than $2i+6$ clean edges where i is the number of long rows plus one from the completely dirty short row.*

Proof. Extended lemma:

1. if the two exterior guards contained in long row i are on back vertices, then long row i contains no more than $2i+6$ clean edges.
2. if the two exterior guards contained in long row i are on fore vertices, then long row i contains no more than $2i+4$ clean edges.
3. if one exterior guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i+5$ clean edges.

In all the above cases, the number of clean edges is less than or equal to $2i+6$, thus, the extended lemma implies the lemma.

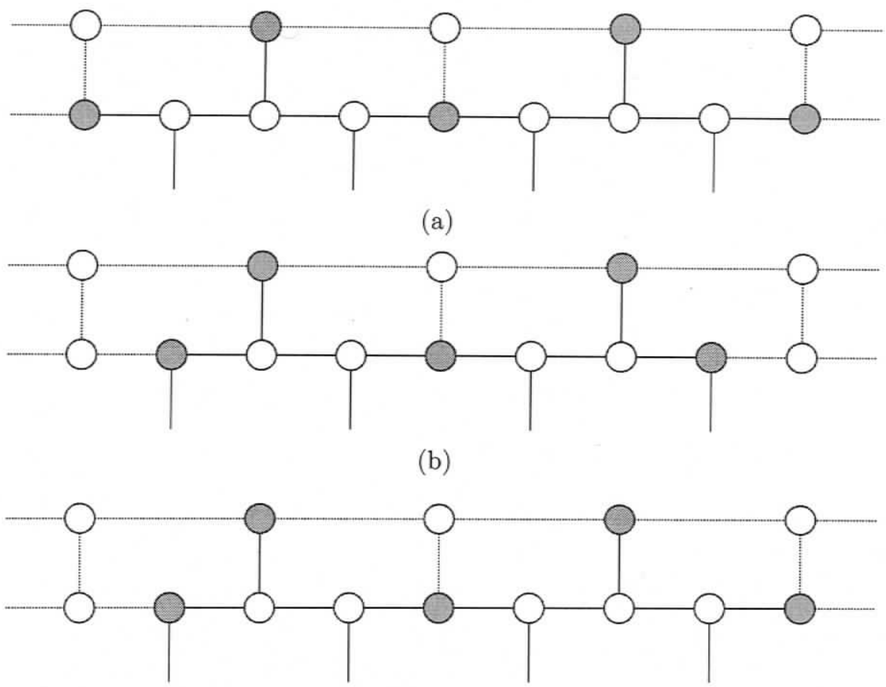
We will prove that any long row i cannot have more than $2i+6$ clean edges by induction on i.

Let long row k be the one with that contains three guards.

Base Case. There are two possible cases for the base case. Either $k=1$, meaning the long row with three guards is adjacent to the completely dirty short row or $k>1$ meaning that the long row with three guards is adjacent only to long rows containing two guards.

Case $i = k = 1$

If the completely dirty short row with two guards is adjacent to one of the long rows with three guards then it is easy to see by Figure 3.13 that the extended lemma holds.



(a)

(b)

(c)

Figure 3.13: Because $i = 1$ the extended lemma holds for 3.13(a) both exterior guards are on back vertices, 3.13(b) both exterior guards on fore vertices and 3.13(c) one exterior guard is on a back vertex and one exterior guard is on a fore vertex.

Case $i = k > 1$

If the completely dirty short row with two guards is adjacent to one of the long rows with two guards then by Lemma 12 and Lemma 9 the maximum number of clean edges is $2i+4$, and thus the base case holds for all items in the extended lemma.

Induction. After long row 1 all remaining long rows have 2 guards and thus the inductive step is essentially the same as the inductive step of Lemma 8. □

Corollary 6. *In a wall, if one short row is completely dirty and contains exactly two guards, one long row contain exactly three guards and all long rows contain exactly two guards, then no long row can contain more than $2h + 2$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 14 provides an upper bound of $2h + 2$ as the most clean edges that any long row may contain. □

Lemma 15. *In a wall, if one short row is completely dirty and contains exactly three guards, and all long rows contain exactly two guards, then long row i can contain no more than $2i + 2$ clean edges where i is the number of long rows plus one from the completely dirty short row.*

Proof. If the completely dirty short had just one guard we know from Lemma 8 the adjacent long row could have at most 4 clean edges. By Lemma 12 we know that even with three guards, the adjacent long row still has no more than 4 clean edges. Hence, the extra guard in the short row does not effect the number of clean edges in the other rows. Thus, by Lemma 8, long row i can contain at most $2i + 2$ clean edges. □

Corollary 7. *In a wall, if one short row is completely dirty and contains exactly three guards, and all long rows contain exactly two guards, then no long row can contain more than $2h - 2$ clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 15 provides an upper bound of $2h - 2$ as the most clean edges that any long row may contain. □

3.4.3 Completely Clean Short Row Lemmas

Lemma 16. *In a wall, if one short row is completely clean and contains exactly one guard, and all long rows contain exactly two guards, then long row i can contain no* *more than $2i+2$ dirty edges, where a row is numbered i if there are $i-1$ rows between it and the completely dirty short row.*

Proof. Extended lemma:

1. if the two guards contained in long row i are on back vertices, then long row i contains no more than $2i + 2$ dirty edges.

2. if the two guards contained in long row i are on fore vertices, then long row i contains no more than $2i$ dirty edges.

3. if one guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i + 1$ dirty edges.

In all the above cases, the number of dirty edges is less than or equal to $2i + 2$, thus, the extended lemma implies the lemma.

We will prove that any long row i cannot have more than $2i + 2$ clean edges by induction on i.

Base Case. $i = 1$

From Figure 3.14 it is easy to see that the extended lemma holds for all three cases.

Induction. Let us assume that the extended lemma is true for all i, $1 \leq i \leq i_0$, for some $i_0 \geq 1$. We show that it is then true for $i_0 + 1$, and hence true for all i.

Case 1: *Suppose both guards contained in long row i_0 are on back vertices*

By the inductive hypothesis, long row i_0 contains no more than $2i_0 + 2$ dirty edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

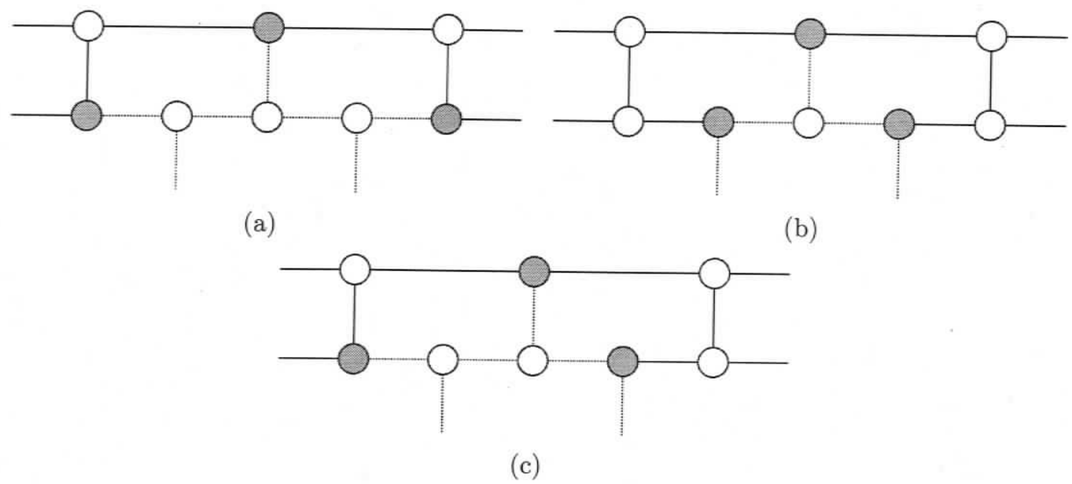
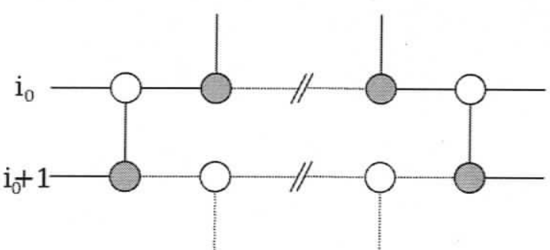
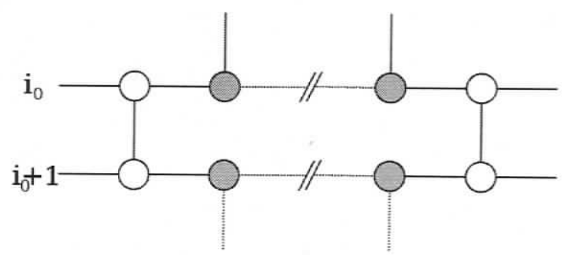


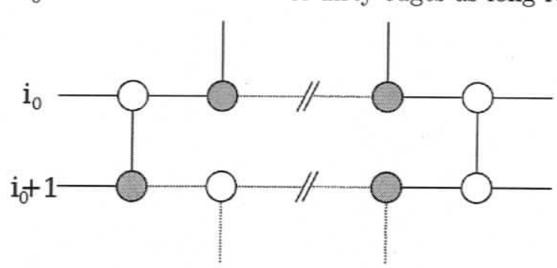
Figure 3.14: The extended lemma holds for 3.14(a) both guards are on back vertices, 3.14(b) both guards on fore vertices and 3.14(c) one guard on a back vertex and one guard on a fore vertex.



(a) Long row i_0+1 can contain at most two more dirty edges than long row i_0.



(b) Long row i_0+1 can contains the same number of dirty edges as long row i_0.



(c) Long row i_0+1 can contain at most one more dirty edge than long row i_0.

Figure 3.15: The various configuration of guards in Case 1

Case 1.1 *Both guards contained in long row* $i_0 + 1$ *are on back vertices*

From Figure 3.15(a) we see that no more than 2 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ dirty edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

Case 1.2 *Both guards contained in long row* $i_0 + 1$ *are on fore vertices*

From Figure 3.15(b) we see that no dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ dirty edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.

Case 1.3 *One guard contained in long row* $i_0 + 1$ *is on a back vertex and the other guard is on a fore vertex*

From Figure 3.15(c) we see that no more than 1 dirty edge can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ dirty edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma.

Case 2: *Suppose both guards contained in long row* i_0 *are on fore vertices*

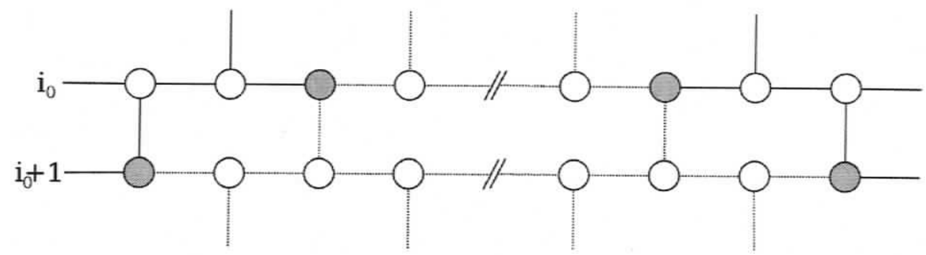
By the inductive hypothesis, long row i_0 contains no more than $2i_0$ dirty edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

Case 2.1 *Both guards contained in long row* $i_0 + 1$ *are on back vertices*

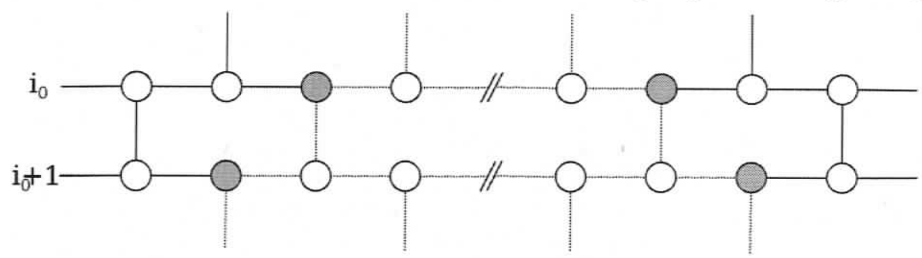
From Figure 3.16(a) we see that no more than 4 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ dirty edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

Case 2.2 *Both guards contained in long row* $i_0 + 1$ *are on fore vertices*

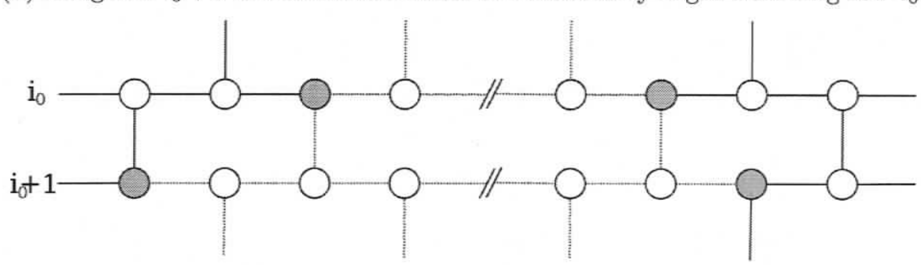
From Figure 3.16(b) we see that no more than 2 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ dirty edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.



(a) Long row $i_0 + 1$ can contain at most four more dirty edges than long row i_0.



(b) Long row $i_0 + 1$ can contains at most two more dirty edges than long row i_0.



(c) Long row $i_0 + 1$ can contain at most three more dirty edges than long row i_0.

Figure 3.16: The various configuration of guards in Case 2

Case 2.3 *One guard contained in long row $i_0 + 1$ is on a back vertex and the other guard is on a fore vertex*

From Figure 3.16(c) we see that no more than 3 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ dirty edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma.

Case 3: *Suppose one guard contained in long row i_0 is on a back vertex and one guard contained in long row i_0 is on a fore vertex*

By the inductive hypothesis, long row i_0 contains $2i_0 + 1$ dirty edges. Consider the possible configurations of guards contained in long row $i_0 + 1$.

Case 3.1 *Both guards contained in long row $i_0 + 1$ are on back vertices*

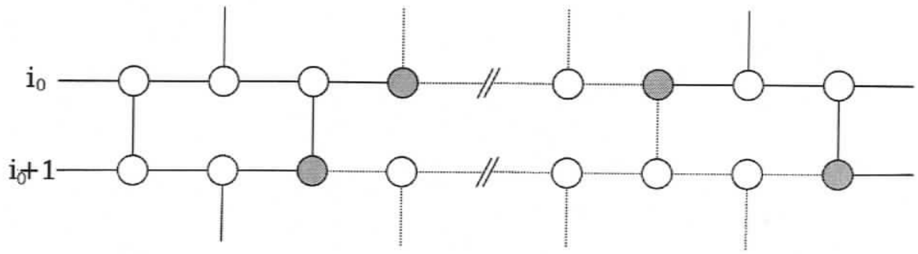
From Figure 3.17(a) we see that no more than 3 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 4$ dirty edges, i.e., no more than $2(i_0 + 1) + 2$, conforming to item 1 in the extended lemma.

Case 3.2 *Both guards contained in long row $i_0 + 1$ are on fore vertices*

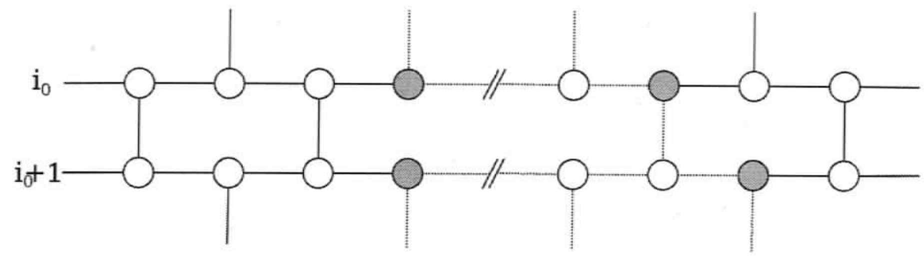
From Figure 3.17(b) we see that no more than 1 dirty edge can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 2$ dirty edges, i.e., no more than $2(i_0 + 1)$, conforming to item 2 in the extended lemma.

Case 3.3 *One guard contained in long row $i_0 + 1$ is on a back vertex and the other guard is on a fore vertex*

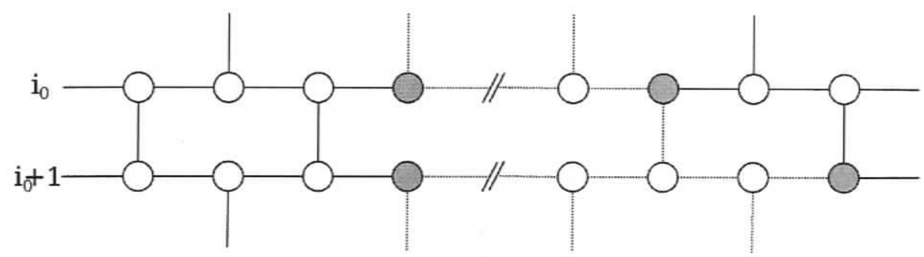
From Figure 3.17(c) and Figure 3.17(d) we see that no more than 2 dirty edges can be gained. Hence, long row $i_0 + 1$ contains no more than $2i_0 + 3$ dirty edges, i.e., no more than $2(i_0 + 1) + 1$, conforming to item 3 in the extended lemma. □



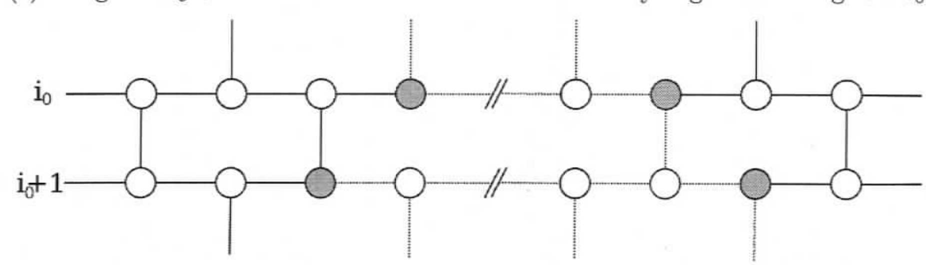
(a) Long row $i_0 + 1$ can contain at most three more dirty edges than long row i_0.



(b) Long row $i_0 + 1$ can contains at most one more dirty edge than long row i_0.



(c) Long row $i_0 + 1$ can contain at most two more dirty edges than long row i_0.



(d) Long row $i_0 + 1$ can contain at most two more dirty edges than long row i_0.

Figure 3.17: The various configuration of guards in Case 3

Corollary 8. *In a wall, if one short row is completely clean and contains exactly one guard, and all long rows contain exactly two guards, then no long row can contain more than $2h - 2$ dirty edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 16 provides an upper bound of $2h - 2$ as the most dirty edges that any long row may contain. □

Lemma 17. *In a wall, if one short row is completely clean and contains exactly one guard, and one long row contains exactly three guards, and all other long rows contain exactly two guards, then long row i can contain no more than $2i+4$ dirty edges, where a row is numbered i if there are $i - 1$ rows between it and the completely dirty short row.*

Proof. Extended lemma:

1. if the two exterior guards contained in long row i are on back vertices, then long row i contains no more than $2i + 4$ dirty edges.

2. if the two exterior guards contained in long row i are on fore vertices, then long row i contains no more than $2i + 1$ dirty edges.

3. if one exterior guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i + 3$ dirty edges.

In all the above cases, the number of dirty edges is less than or equal to $2i + 4$, thus, the extended lemma implies the lemma.

We will prove that any long row i cannot have more than $2i + 4$ dirty edges by induction on i.

Let long row k be the one that contains three guards.

Base Case. There are two possible cases for the base case. Either $k = 1$, meaning the long row with three guards is adjacent to the completely clean short row or $k > 1$

meaning that the long row with three guards is adjacent only to long rows containing two guards.

Case $i = k = 1$

From Figure 3.8 it is easy to see that the extended lemma holds for all three cases.

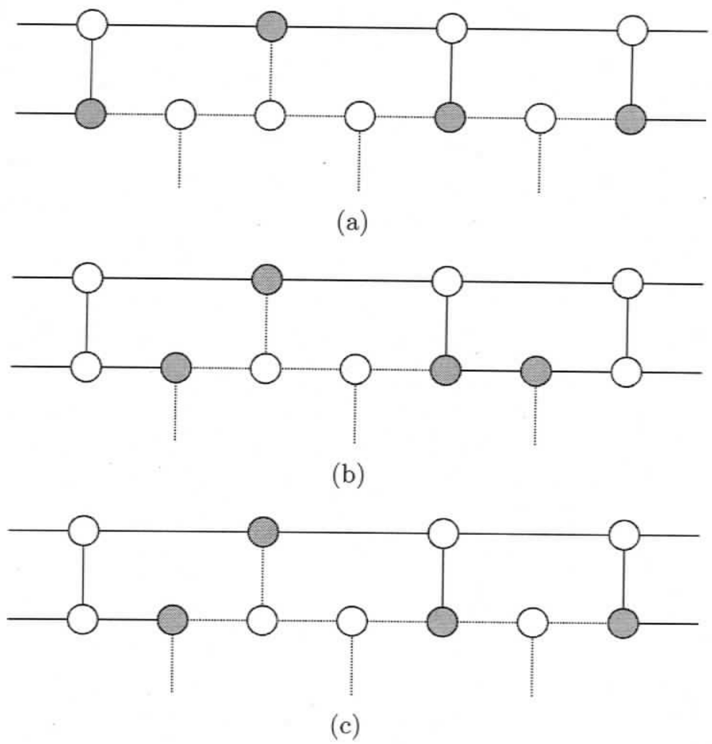


Figure 3.18: Because $i = k = 1$ the extended lemma holds for 3.18(a) both exterior guards are on back vertices, 3.18(b) both exterior guards on fore vertices and 3.18(c) one exterior guard is on a back vertex and one exterior guard is on a fore vertex.

Case $i = k > 1$

The guards in long row $i - 1$ can be either on back vertices, fore vertices and on both a back and a fore vertex.

Consider the following three cases:

Case 1. *Both guards on long row $i-1$ are on back vertices*

From Lemma 16's extended lemma item 1 we know that there cannot be more than $2(i-1)+2 = 2i$ clean edges on long row $i-1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.

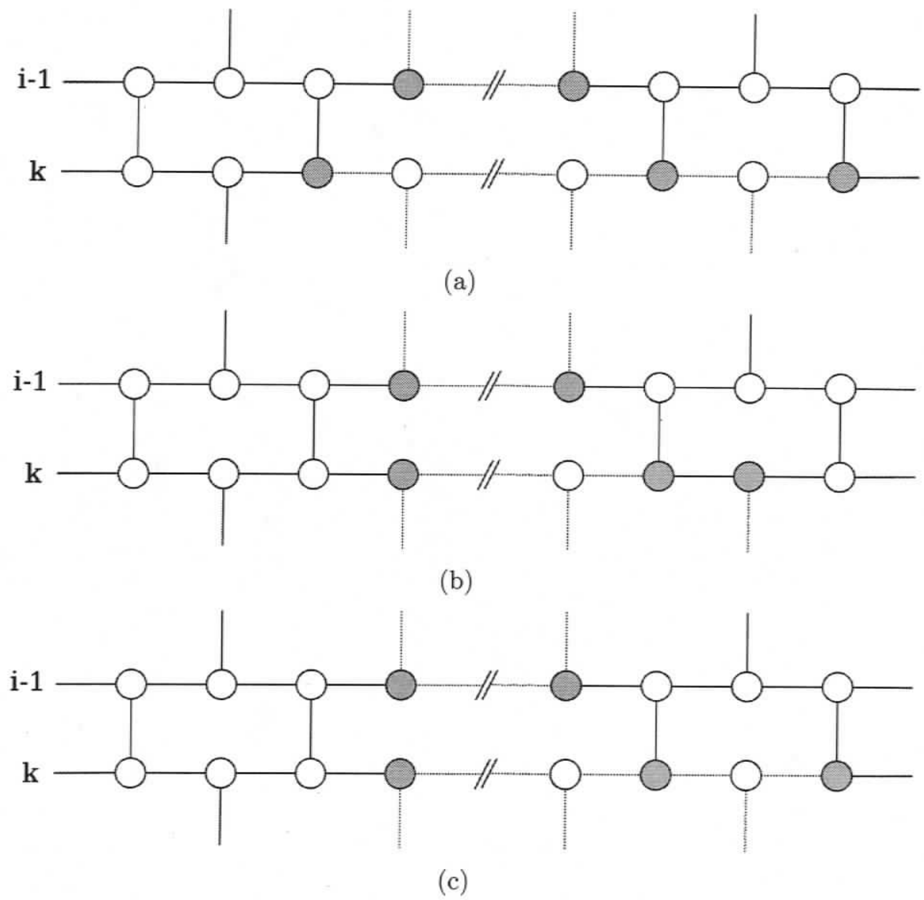


Figure 3.19: The various configuration of guards in Case 1

Case 1.1 *Both exterior guards on long row k are on back vertices.*

From Figure 3.19(a) we see that no more than 4 dirty edges can be gained. Hence, long row i contains no more than $2i+4$ dirty edges, conforming to item 1 in the extended lemma.

Case 1.2 *Both exterior guards guards on long row k are on fore vertices.*

From Figure 3.19(b) we see that no more than 1 dirty edges can be gained. Hence, long row i contains no more than $2i + 1$ dirty edges, conforming to item 2 in the extended lemma.

Case 1.3 *One exterior guard on long row k is on a back vertex, the other is on a fore vertex.*

From Figure 3.19(c) we see that no more than 3 dirty edges can be gained. Hence, long row i contains no more than $2i + 3$ dirty edges, conforming to item 3 in the extended lemma.

Case 2. *Both guards on long row* $i - 1$ *are on fore vertices*

From Lemma 16's extended lemma item 2 we know that there cannot be more than $2(i-1) = 2i-2$ dirty edges on long row $i-1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.

Case 2.1 *Both exterior guards on long row k are on back vertices.*

From Figure 3.20(a) we see that no more than 6 dirty edges can be gained. Hence, long row i contains no more than $2i + 4$ dirty edges, conforming to item 1 in the extended lemma.

Case 2.2 *Both exterior guards guards on long row k are on fore vertices.*

From Figure 3.20(b) we see that no more than 3 dirty edges can be gained. Hence, long row i contains no more than $2i + 1$ dirty edges, conforming to item 2 in the extended lemma.

Case 2.3 *One exterior guard on long row k is on a back vertex, the other is on a fore vertex.*

From Figure 3.20(c) we see that no more than 5 dirty edges can be

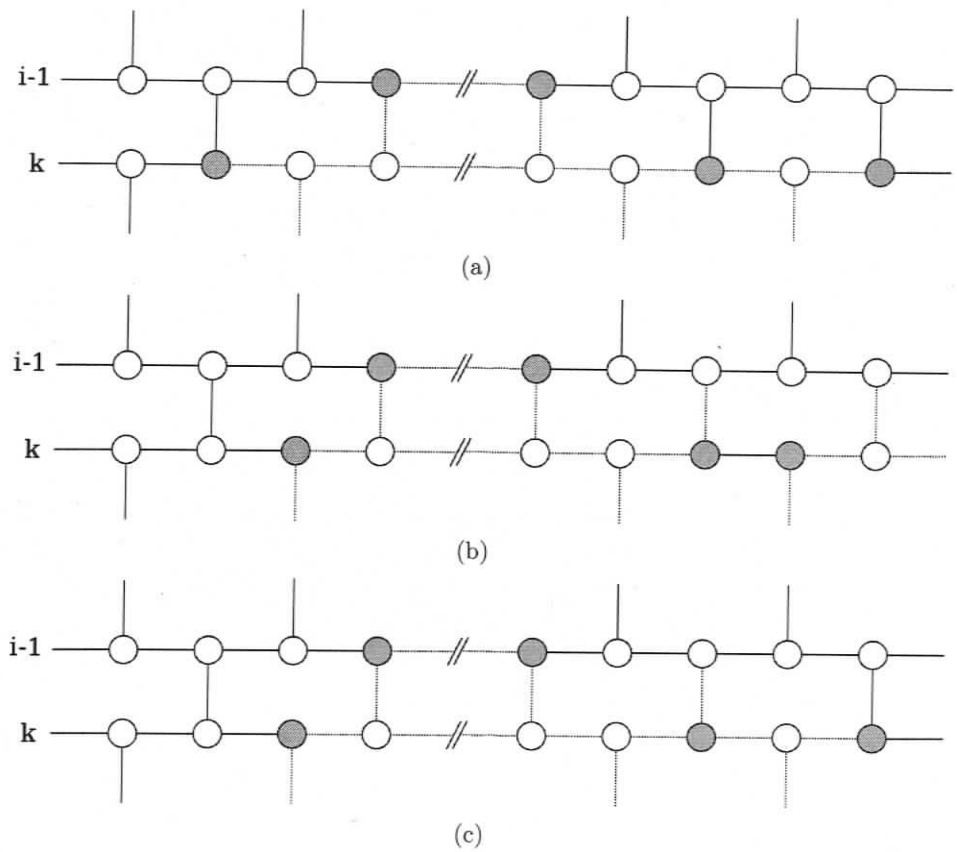
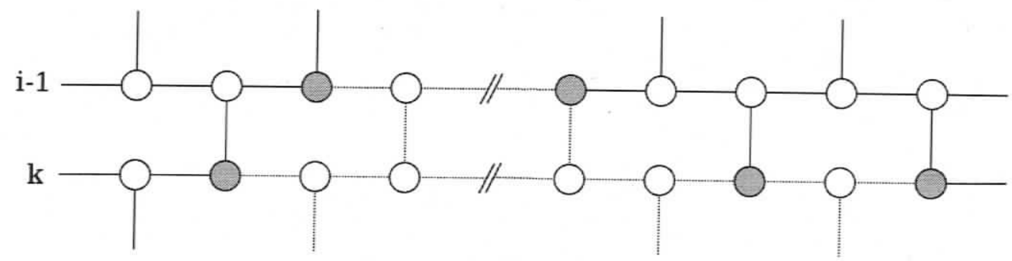


Figure 3.20: The various configuration of guards in Case 2

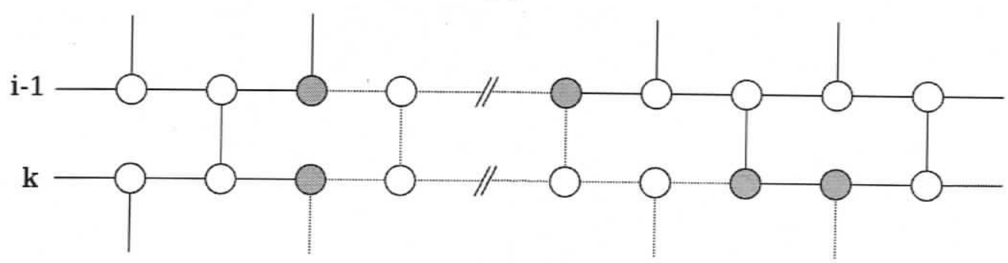
gained. Hence, long row i contains no more than $2i + 3$ dirty edges, conforming to item 3 in the extended lemma.

Case 3. *One guard on long row* $i - 1$ *is on a back vertex, the other is on a fore vertex*

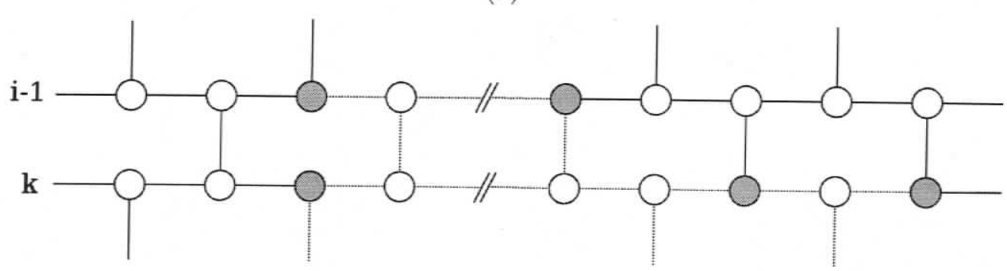
From Lemma 16's extended lemma item 3 we know that there cannot be more than $2(i-1) + 1 = 2i - 1$ dirty edges on long row $i - 1$. Of guards in long row k: both exterior guards can be on back vertices, both exterior guards can be on a fore vertex or one exterior guard can be on a fore vertex and the other on a back vertex.



(a)



(b)



(c)

Figure 3.21: The various configuration of guards in Case 3

Case 3.1 *Both exterior guards on long row* k *are on back vertices.*

From Figure 3.21(a) we see that no more than 5 dirty edges can be gained. Hence, long row i contains no more than $2i + 4$ dirty edges, conforming to item 1 in the extended lemma.

Case 3.2 Both exterior guards guards on long row k are on fore vertices.

From Figure 3.21(c) we see that no more than 2 dirty edges can be gained. Hence, long row i contains no more than $2i + 1$ dirty edges, conforming to item 2 in the extended lemma.

Case 3.3 One exterior guard on long row k is on a back vertex, the other is on a fore vertex.

From Figure 3.21(b) we see that no more than 4 dirty edges can be gained. Hence, long row i contains no more than $2i + 3$ dirty edges, conforming to item 3 in the extended lemma.

Induction. After long row k all remaining long rows have 2 guards and thus the inductive step is essentially the same as the inductive step of Lemma 8. □

Corollary 9. *In a wall, if one short row is completely clean and contains exactly one guard, and one long row contains exactly three guards, and all other long rows contain exactly two guards, then no long row can contain more than $2h$ dirty edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 17 provides an upper bound of $2h$ as the most dirty edges that any long row can contain. □

Lemma 18. *In a wall, if one short row is completely clean and contains exactly two guards, and all long rows contain exactly two guards, then long row i can contain no more than $2i+4$ dirty edges, where a row is numbered i if there are $i-1$ rows between it and the completely dirty short row.*

Proof. With the extra guard on the short row, it is possible to guard four connections between the rows, using the two guards in the long row and two guards in the short row. This means that the long row 1 adjacent to the short row can contain up to six dirty edges. A configuration of guards that allows six dirty edges in long row 1 is illustrated in Figure 3.22.

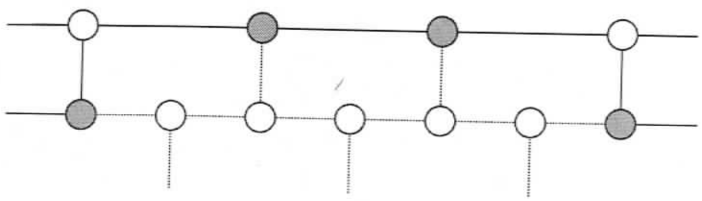


Figure 3.22: Long row 1 can contain no more than six dirty edges.

As in Lemma 16, each long row increases the number of dirty edges by no more than two. Hence, we can have 2 more dirty edges than in Lemma 16 so a long row i may have at most $2i + 4$ dirty edges. $\square$

Corollary 10. *In a wall, if one short row is completely clean and contains exactly two guards, and all long rows contain exactly two guards, then long row i can contain no more than $2i + 4$ dirty edges, where a row is numbered i if there are $i - 1$ rows between it and the completely dirty short row.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 18 provides an upper bound of $2h$ as the most dirty edges that any long row can contain. $\square$

Lemma 19. *In a wall, if one short row is completely clean and contains no guards, and all long rows are partly dirty and contain exactly two guards, then no long row can contain more than $2i$ dirty edges.*

Proof. Extended lemma:

1. if the two guards contained in long row i are on back vertices, then long row i contains no more than $2i$ dirty edges.

2. if the two guards contained in long row i are on fore vertices, then long row i contains no more than $2i-2$ dirty edges.

3. if one guard contained in long row i is on a fore vertex and the other on a back vertex then long row i contains no more than $2i-1$ dirty edges.

In all the above cases, the number of dirty edges is less than or equal to $2i$, thus, the extended lemma implies the lemma.

We will prove that any long row i cannot have more than $2i$ clean edges by induction on i.

Base Case. $i = 1$

For the remaining case, from Figure 3.23, it is easy to see that the extended lemma holds for items 1 and 2. As for item 3, since $0 \leq 1$ the extended lemma holds.

Induction. The inductive step is effectively identical to that of Lemma 16. The only change is that since the base case had two less clean edges meaning that all the results from Lemma 16 are reduced by two. Hence, there are at most $2i_0$ for some $i_0 > 1$. □

Corollary 11. *In a wall, if one short row is completely clean and contains no guards, and all long rows contain exactly two guards, then no long row can contain more than* $2h-4$ *dirty edges.*

Proof. There are $h-2$ long rows. Setting i to $h-2$ in Lemma 19 provides an upper bound of $2h-4$ as the most dirty edges that any long row can contain. □

Lemma 20. *In a wall, if one short row is completely clean and contains no guards, one long row contains three guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i+2$ *dirty edges.*

Proof. This proof is essentially the same as Lemma 17 except that instead of citing Lemma 16 one needs to cite Lemma 19. Additionally, the base case where $k = 1$ is slightly different but still easy to see. □

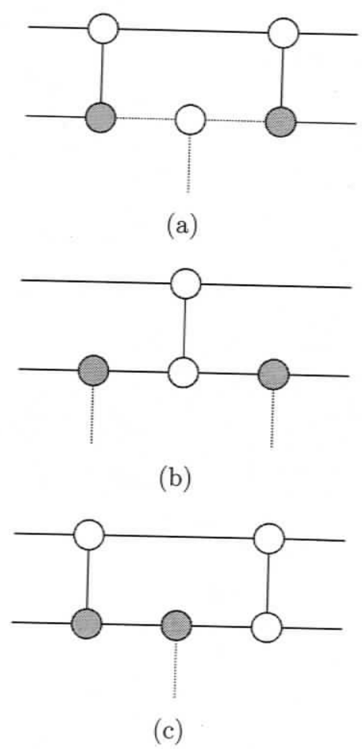


Figure 3.23: Because $i = k = 1$ the extended lemma holds for 3.23(a) both guards are on back vertices, 3.23(b) both guards on fore vertices and 3.23(c) one guard is on a back vertex and one guard is on a fore vertex.

Corollary 12. *In a wall, if one short row is completely clean and contains no guards, one long row contains three guards and all other long rows contain exactly two guards, then no long row can contain more than* $2h - 2$ *dirty edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 20 provides an upper bound of $2h - 2$ as the most dirty edges that any long row can contain. □

Lemma 21. *In a wall, if one short row is completely clean and contains no guards, one long row contains four guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i + 4$ *dirty edges.*

Proof. This proof follows that same pattern as Lemma 20 except instead of a long row with three guards there is a long row with four guards. □

Corollary 13. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than* $2h$ *clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 21 provides an upper bound of $2h$ as the most dirty edges that any long row can contain. □

Lemma 22. *In a wall, if one short row is completely clean and contains no guards, one long row contains four guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i + 6$ *dirty edges.*

Proof. This proof follows that same pattern as Lemma 20 except instead of a long row with three guards there is a long row with five guards. □

Corollary 14. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than* $2h + 2$ *clean edges.*

Proof. There are $h-2$ long rows. Setting i to $h-2$ in Lemma 22 provides an upper bound of $2h+2$ as the most dirty edges that any long row can contain. □

Lemma 23. *In a wall, if one short row is completely clean and contains no guards, one long row contains four guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i+6$ *dirty edges.*

Proof. This proof follows that same pattern as Lemma 20 except let k be the long row that contains four guards and instead of citing Lemma 19 one needs to cite Lemma 20. □

Corollary 15. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than* $2h+2$ *clean edges.*

Proof. There are $h-2$ long rows. Setting i to $h-2$ in Lemma 23 provides an upper bound of $2h+2$ as the most dirty edges that any long row can contain. □

Lemma 24. *In a wall, if one short row is completely clean and contains no guards, one long row contains four guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i+4$ *dirty edges.*

Proof. This proof follows that same pattern as Lemma 20 except let k be the second long row that contains three guards and instead of citing Lemma 19 one needs to cite Lemma 17. □

Corollary 16. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than* $2h$ *clean edges.*

Proof. There are $h-2$ long rows. Setting i to $h-2$ in Lemma 24 provides an upper bound of $2h$ as the most dirty edges that any long row can contain. □

Lemma 25. *In a wall, if one short row is completely clean and contains no guards, one long row contains four guards and all other long rows contain exactly two guards, then no long row can contain more than* $2i + 6$ *dirty edges.*

Proof. This proof follows that same pattern as Lemma 25 except let k be the third long row that contains three guards and instead of citing Lemma 20 one needs to cite Lemma 24. □

Corollary 17. *In a wall, if one short row is completely dirty and contains exactly one guard, and one long row contains exactly four guards, and all other long rows contain exactly two guards, then no long row can contain more than* $2h + 2$ *clean edges.*

Proof. There are $h - 2$ long rows. Setting i to $h - 2$ in Lemma 25 provides an upper bound of $2h + 2$ as the most dirty edges that any long row can contain. □

3.4.4 Tower Lemmas

Lemma 26. *Consider a wall where one of the short rows is partly dirty, the other short row is completely clean, all long rows are partly dirty and at most one long row has three guards and all other long rows have two guards. If the partly dirty short row is adjacent to a long row with two guards and there exists a tower of height* $h - 2$ *and width* 4*, then there exist at least two clean columns.*

Proof. All the edges connecting tower vertices in adjacent long rows must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Hence it remains to show that at least two clean edges exist connecting each of the top and bottom rows of the tower to the short rows and that these edges are in the right places to create two completely clean columns.

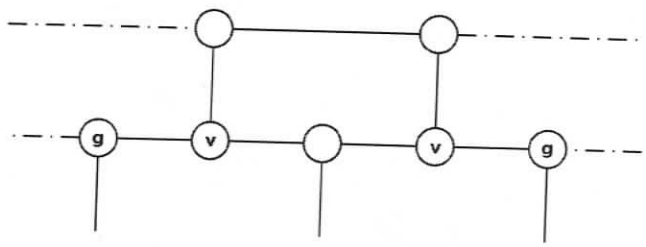


Figure 3.24: In the event of two edges connecting the tower and the completely dirty short row, both must be clean or recontamination will occur.

We note that the five vertices in the top and bottom rows of the tower are each adjacent to either 3 or 2 vertices in the short rows.

Case partly dirty short row

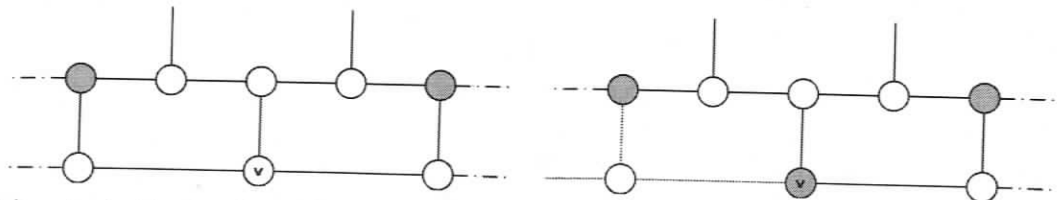
Consider the following two cases:

Case there are two edges connecting the tower and partly dirty short row

There must be at least four clean edges in the long rows because of the tower. But since there are only two guards in the long row adjacent to the partly dirty short row those guards must be either on the vertices labeled g in Figure 3.24 or not in the tower. In either case, they are not on the vertices labeled v in Figure 3.24 which is where they are needed to prevent recontamination from the short row. Therefore, the two connecting edges must be clean.

Case there are three edges connecting the tower and partly dirty short row

Since there are only two guards in the long row adjacent to the partly dirty short row the tower will be recontaminated unless vertex v (see Figure 3.25) in the partly dirty short row is either incident with only clean edges or guarded.



(a) v is incident only with clean edges so all three edges connecting the tower and the partly dirty short row must be clean.

(b) v is guarded so it must be incident with exactly one clean edge in the partly dirty short row and so two edges connecting the tower and partly dirty short row must be clean.

Figure 3.25: When there are three edges connecting the tower to the partly dirty short row, recontamination can occur unless certain conditions in the partly dirty short row are met.

If it is not guarded then both other connecting edges must also be clean, see Figure 3.25(a).

If it is guarded, then because the short row is partly dirty it must still be incident with at least one clean edge in the short row, see Figure 3.25(b). In this case only two of the connecting edges will be clean, but one of them must always be the connecting edge incident with v.

Case completely clean short row

All edges connecting tower vertices in a long row with adjacent short row vertices must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Thus, in all the above cases, there are at least two clean edges. Furthermore, these edges are no more than one edge apart.

Now consider the two cases, h even and h odd.

Case h is even

By the definition of walls, if h is even then the connections between both short rows and their adjacent long rows occur on even vertices. Thus, both short rows are either connected with 3 edges or 2 edges and these edges correspond.

If the short rows are connected with 2 edges then it is easy to see that regardless of which definition of a column is used the edges that are both column and row edges will always be clean, see Figure 3.26.

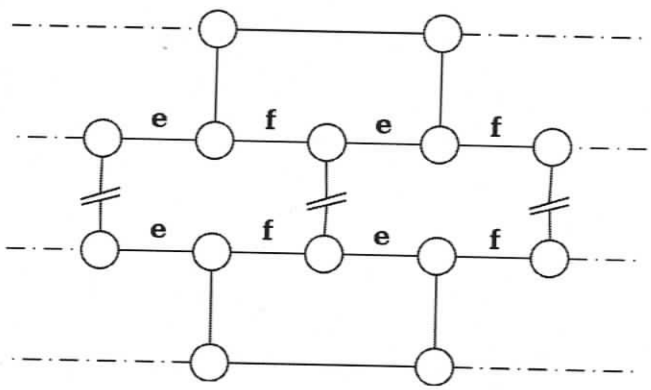


Figure 3.26: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. In either case, the edges must be clean.

If the short rows are connected with 3 edges then the three edges connected to the completely clean short row must be clean but only two of those edges connecting to the partly dirty short row need to be clean.

However, if there are only two clean edges connecting to the partly dirty short row they must be either a and b or b and c from Figure 3.27. Similarly, there are two definitions of columns, one of which will use both a and b and the other which will use both b and c. Thus, under at least one definition there will be two columns.

Case h is odd

By the definition of walls, if h is odd then the connections between one of the short rows and its adjacent long row occurs on even vertices while the other

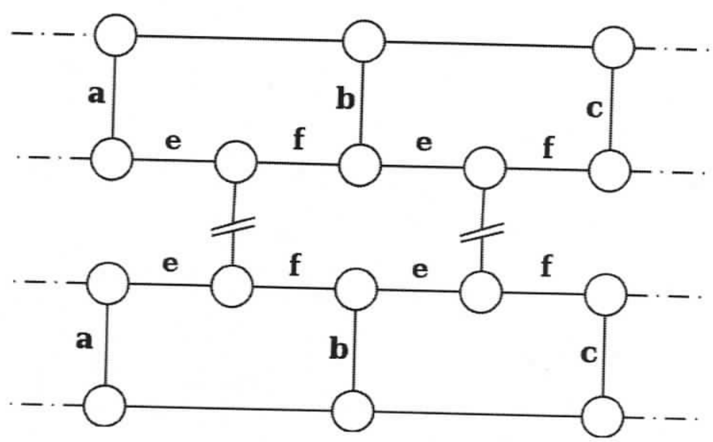


Figure 3.27: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. So either a column is formed with edges labeled a,b and e or a column is formed using the edges labeled b,c and f.

occurs on odd vertices. Furthermore, this means that if one short row has a 2 edge connection with the long row then the other must have a 3 edge connection.

If the short row with 3 edge connections is the completely clean short row then all edge connections between both short rows will be clean and thus there must be two clean columns, see Figure 3.28.

If the short row with 3 edge connections is the partly dirty short row then it is possible that either edge a or c (see Figure 3.28) could be dirty. However, one of the definitions for a column will provide two clean columns, see Figure 3.28.

In either case there must exist two clean columns. □

Lemma 27. *Consider a wall where one of the short rows is partly dirty, the other short row is completely clean, all long rows are partly dirty and one long row has three guards and all other long rows have two guards. If the partly dirty short row is adjacent to a long row with three guards and there exists a tower of height $h-2$ and width 6, then there exist at least two clean columns.*

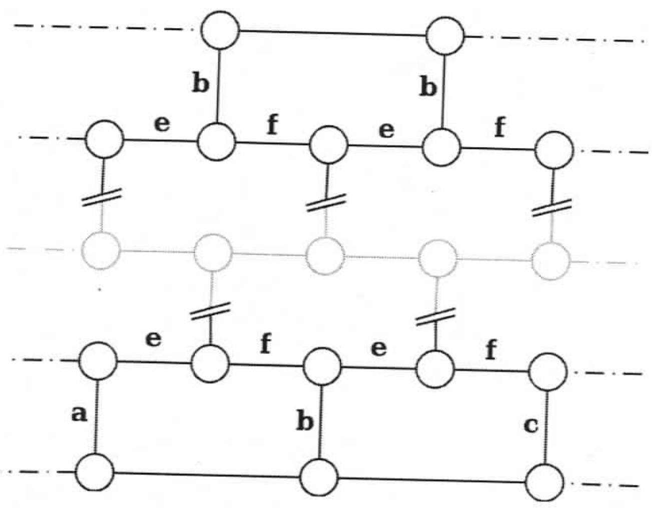


Figure 3.28: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. So either a column is formed with edges labeled a,b and e or a column is formed using the edges labeled b,c and f.

Proof. All the edges connecting tower vertices in adjacent long rows must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Hence it remains to show that at least two clean edges exist connecting each of the top and bottom rows of the tower to the short rows and that these edges are in the right places to create two completely clean columns.

We note that the seven vertices in the top and bottom rows of the tower are each adjacent to either 4 or 3 vertices in the short rows.

Case partly dirty short row

Consider the following two cases:

Case there are three edges connecting the tower and partly dirty short row

There must be at least six clean edges in the long rows because of the tower. But since there are only two guards in the long row adjacent to the

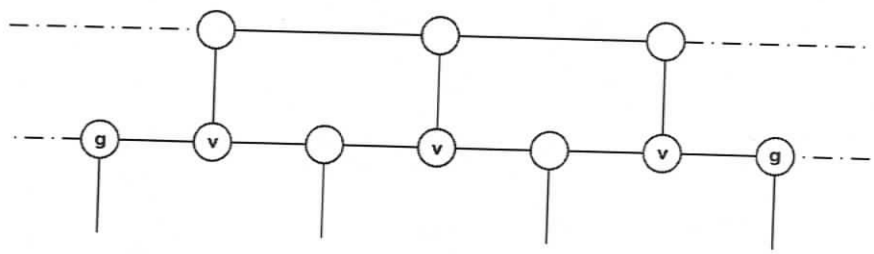


Figure 3.29: In the event of three edges connecting the tower and the completely dirty short row, both must be clean or recontamination will occur.

partly dirty short row those guards must be either on the vertices labeled g in Figure 3.24 or not in the tower. In either case, they are not on the vertices labeled v in Figure 3.25 which is where they are needed to prevent recontamination from the short row. Therefore, the two connecting edges must be clean.

Case there are three edges connecting the tower and partly dirty short row

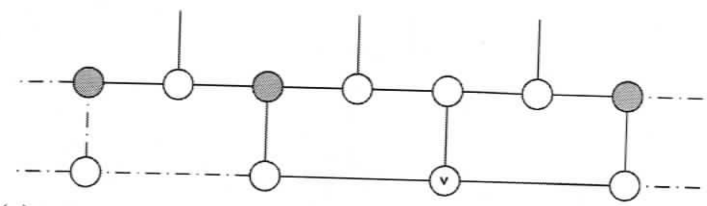
Since there are only three guards in the long row adjacent to the partly dirty short row, one can recontaminate the tower unless vertex v (see Figure 3.30) in the partly dirty short row is either incident with only clean edges or guarded.

If it is not guarded then both other connecting edges must also be clean, see Figure 3.30(a).

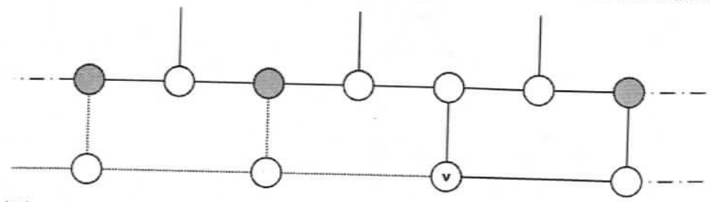
If it is guarded, then because the short row is partly dirty it must still be incident with at least one clean edge in the short row, see Figure 3.30(b). In this case only two of the connecting edges will be clean, but one of them must always be the connecting edge incident with v.

Case completely clean short row

All edges connecting tower vertices in a long row with adjacent short row vertices



(a) v is incident only with clean edges so all three edges connecting the tower and the partly dirty short row must be clean.



(b) v is guarded so it must be incident with exactly one clean edge in the partly dirty short row and so two edges connecting the tower and partly dirty short row must be clean.

Figure 3.30: When there are four edges connecting the tower to the partly dirty short row, recontamination can occur unless certain conditions in the partly dirty short row are met.

must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Thus, in all the above cases, there are at least two clean edges. Furthermore, these edges are no more than one edge apart.

Now consider the two cases, h even and h odd.

Case h is even

By the definition of walls, if h is even then the connections between both short rows and their adjacent long rows occur on even vertices. Thus, both short rows are either connected with 4 edges or 3 edges and these edges correspond.

If the short rows are connected with 3 edges then it is easy to see that regardless of which definition of a column is used the edges that are both column and row edges will always be clean, see Figure 3.31.

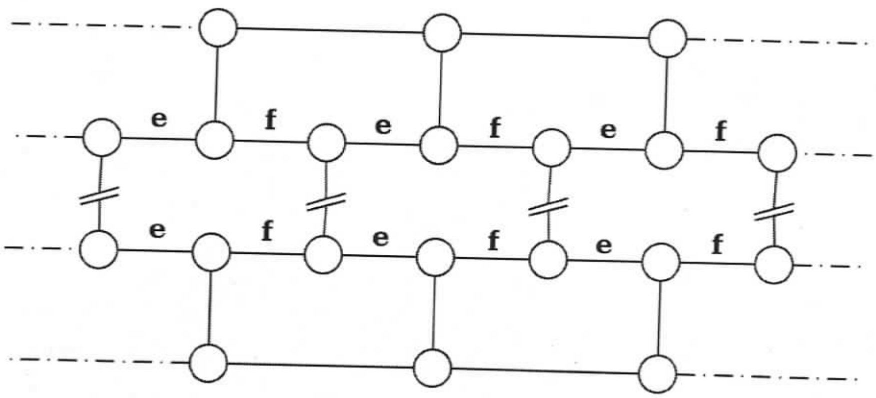


Figure 3.31: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. In either case, the edges must be clean.

If the short rows are connected with 4 edges then the three edges connected to the completely clean short row must be clean but only two of those edges connecting to the partly dirty short row need to be clean.

However, if there are only two clean edges connecting to the partly dirty short row they must be either a and b or b and c from Figure 3.32. Similarly, there are two definitions of columns, one of which will use both a and b and the other which will use both b and c. Thus, under at least one definition there will be two columns.

Case h is odd

By the definition of walls, if h is odd then the connections between one of the short rows and its adjacent long row occurs on even vertices while the other occurs on odd vertices. Furthermore, this means that if one short row has a 3 edge connection with the long row then the other must have a 4 edge connection.

If the short row with 4 edge connections is the completely clean short row then all edge connections between both short rows will be clean and thus there must be two clean columns, see Figure 3.33.

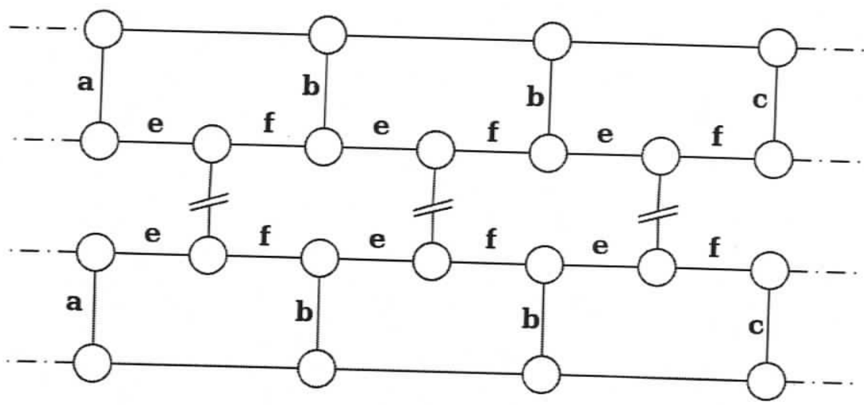


Figure 3.32: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. So either a column is formed with edges labeled a,b and e or a column is formed using the edges labeled b,c and f. No matter what, one of the edges labeled b on both the top and bottom may be clean, however, they are not necessarily both clean.

If the short row with 4 edge connections is the partly dirty short row then it is possible that either edge a or c (see Figure 3.33) could be dirty. However, one of the definitions for a column will provide two clean columns, see Figure 3.33.

In either case there must exist two clean columns. □

Lemma 28. *Consider a wall where both of the short rows are completely clean, all long rows are partly dirty and at most one long row has three guards and all other long rows have two guards. If there exists a tower of height $h-2$ and width 4, then there exist at least two clean columns.*

Proof. All the edges connecting tower vertices in adjacent long rows must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Hence it remains to show that at least two clean edges exist connecting each of the top and bottom rows of the tower to the short rows and that these edges are in

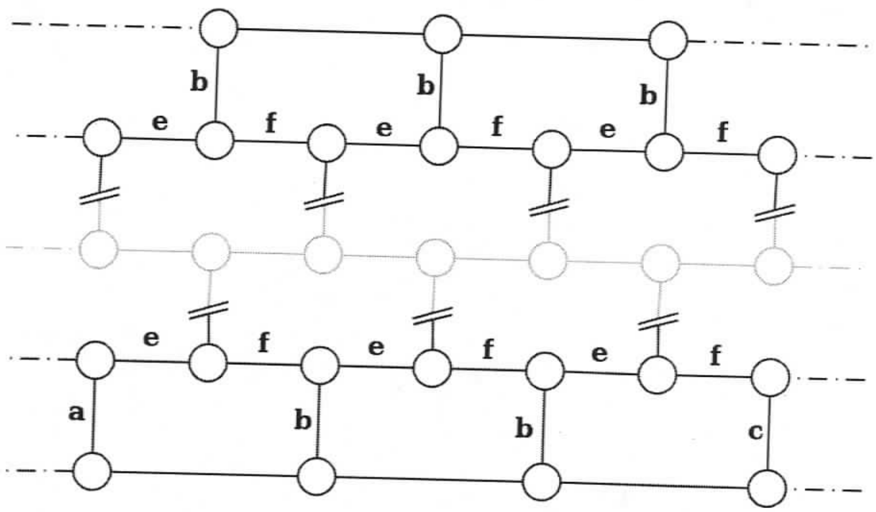


Figure 3.33: Depending on the definition of a column, either all the edges labeled e are in the column or all the edges labeled f are in the column. So either a column is formed with edges labeled a,b and e or a column is formed using the edges labeled b,c and f.

the right places to create two completely clean columns.

We note that the five vertices in the top and bottom rows of the tower are each adjacent to either 3 or 2 vertices in the short rows.

Since both short rows are completely clean all edges connecting tower vertices in a long row with adjacent short row vertices must be clean because suppose one such edge is dirty. Since the row edges which share a vertex with this edge are clean, there must be a guard on both the vertices incident with this dirty edge. But then it must be clean.

Now consider the two cases, h even and h odd.

Case h is even

By the definition of walls, if h is even then the connections between both short rows and their adjacent long rows occur on even vertices. Thus, both short rows are either connected with 3 edges or 2 edges and these edges correspond.

Regardless of the number of connections or the definition of a column used, since all connections are clean it is easy to see that there must exist two clean columns, see Figures 3.26 and 3.27.

Case h is odd

By the definition of walls, if h is odd then the connections between one of the short rows and its adjacent long row occurs on even vertices while the other occurs on odd vertices. Furthermore, this means that if one short row has a 2 edge connection with the long row then the other must have a 3 edge connection. Regardless of the definition of a column used, since all connections are clean it is easy to see that there must exist two clean columns, see Figure 3.28.

In either case there must exist two clean columns. □

3.4.5 Wall Lower Bounds

Lemma 29. *Let G be a wall,* $\mathrm{ns}(G) \geq \min(2h+1, w/2+2)$.

Proof. Let step t be the first step at which the set of clean edges defines a component C^t which includes two columns or one long row (including the wrap-around edge). Step t is a placement step, because it expands the set of clean edges.

There are four distinct cases to consider: C^t includes one long row but no columns, C^t includes one long row and one column, C^t includes one long row and two columns and C^t includes two columns but no long row.

Case 1. *Suppose C^t contains a long row, say R, and no columns.*

Since there are no completely clean columns, every column is either partly or completely dirty at step t and at step $t-1$.

In C^{t-1} there is an almost clean row. This row will have exactly one vertex incident with only dirty edges; let z be that vertex.

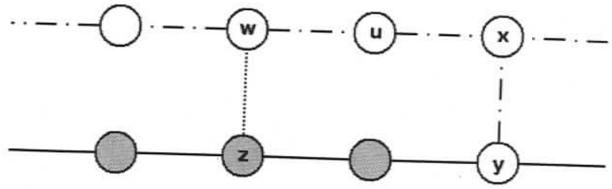


Figure 3.34: Either w, u, x or y must be guarded or else R will be contaminated.

Every column at $t-1$, except the column containing z, is partly dirty since it is not completely clean but it does share an edge (since columns and rows share an edge) with the almost clean row and hence contains at least one clean edge. Thus, in each such column, there is a vertex that is incident with one dirty and one clean edge. Since there are $w/2$ columns, this means that there must be at least $w/2-1$ guards at step $t-1$.

Vertex z is incident with two dirty row edges. Both of these dirty edges must, in addition to being incident with z, be incident with a guarded vertex since the row is almost clean. One of these guards will be in a column other than the one containing z. The other must be in the same column as z since R is a long row. So there are at least $w/2$ guards in $w/2$ columns, at step $t-1$.

By definition of a wall, z is adjacent to a node, call it w, in an adjacent row. Since z is not guarded in C^{t-1} and z is incident with dirty edges, the edge $\{w, z\}$ is not in C^{t-1}, i.e., it is dirty at step $t-1$.

There is a path from w to row R that does not include z. Some vertex along this path must be guarded, otherwise recontamination of R will occur. See Figure 3.34. This means that some column contains a second guard, and hence there are at least $w/2+1$ guards at step $t-1$.

Since t is a placement step, the number of guards is increased. Hence, there must be at least $w/2+2$ guards at step t.

In the remaining three cases, C^{t-1} may contain at least one clean column and will never contain a completely clean long row. In the following cases, when a column is clean, the following is true:

Since every column shares an edge with every long row, and there exists a clean column every long row must be partly clean. Hence, at $t-1$, every long row contains at least two guards.

If a short row is partly dirty, it must contain at least two guards because of the wrap-around edges.

Since in each of the cases there is a completely clean column every completely dirty short row must have at least one guard.

Whenever there are $2h$ critical guards on the graph at step $t-1$ the lemma is proven since step t is a placement, and thus, there will be $2h+1$ guards on the graph. Henceforth we only deal with the situations that are not trivial cases:

Case A. Both short rows contain no guards, and either (a) at most one long row contains at most five guards, (b) at most one long row contains at most four guards and at most long row contains three guards or (c) no more than three long rows contain three guards. All other long rows contain two guards.

Case B. One short row contains no guards and the other short row contains one guard and either at most one long row contains at most four guards or at most two long rows contain three guards. All other long rows contain two guards.

Case C. One short row contains no guards and the other short row contains two guards and at most one long row contains three guards and all other long rows contain two guards.

Case D. One short row contains no guards and the other short row contains three guards and all long rows contain two guards.

Case E. Both short rows contain one guard and at most one long row contains three guards and all other long rows contain two guards.

Case F. One short row contains one guard and the other short row contains two guards and all long rows contain exactly two guards.

Case 2 *Suppose C^t contains a long row, say R, and one column.*

At step $t-1$, there may or may not exist a clean column. If no clean column exists at $t-1$, then the proof is almost identical to Case 1 which only uses the fact that there is no clean column at $t-1$. Otherwise, we suppose there is a clean column and consider the following:

At step $t-1$, R is partly dirty, but it is completely clean at step t. Hence it contains two adjacent dirty edges incident with an unguarded vertex and two guards, at step $t-1$. So R must have exactly $w-2$ clean edges.

We consider six possible cases depending on whether the two short rows are partly dirty, completely clean or completely dirty at step $t-1$. Since the placement at step t is to R, a long row, the number of guards on either short row at step t is unchanged from the number at step $t-1$.

Case 2.1. *At step $t-1$, suppose both short rows are partly dirty.*

Then each short row contains at least two guards, since the rows wrap-around and contain both clean and dirty edges. Hence, there are at least $2h+1$ guards on the graph at step t.

Case 2.2. *At step $t-1$, suppose one short row is partly dirty and one is completely clean.*

Consider the four non-trivial cases:

Case 2.2.1 *The completely clean short row contains one guard, the partly dirty short row contains two guards and all long rows contain two guards.*

From Corollary 8 we know that no long row can have more than $2h-2$ dirty edges. Therefore, there must be at least $w-(2h-2)$ clean edges in any long row.

Furthermore, from Lemma 16 we know that any long row j where $1 \leq j < i$ has less dirty edges than in long row i and, from that proof we know that in order to achieve the maximum number of clean edges in each long row, a "pyramid" of clean edges is required. Finally, because there are at most 2 guards per long row, the clean component must be continuous within each long row.

Thus, there must exist a tower of width $w-(2h-2)$.

Since there is only one clean column, from Lemma 26 there is no tower of width 4. Hence $w-(2h-2) < 4$.

Therefore $w < 2h+2$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+2}{2}+2 < h+3 \leq h+2$ guards. The current strategy uses $2h$ guards and $2h \geq h+2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.2.2 *The completely clean short row contains no guards, the partly dirty short row contains two guards and all long rows contain two guards.*

From Corollary 11 we know that no long row can have more than $2h-4$ dirty edges. Therefore, there must be at least $w-(2h-4)$ clean edges in any long row.

As in explained in Case 2.2.1, the clean component must be continuous.

Thus, there must exist a tower of width $w - (2h - 4)$.

Since there is only one clean column, from Lemma 26 there is no tower of width 4. Hence $w - (2h - 4) < 4$.

Therefore $w < 2h$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h}{2} + 2 < h + 2 \leq h + 1$ guards. The current strategy uses $2h - 1$ guards and $2h - 1 \geq h + 1$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.2.3 *The completely clean short row contains no guards, the partly dirty short row contains two guards, one long row contains three guards and all long rows contain two guards. The partly dirty short row is adjacent to a long row containing two guards.*

From Corollary 12 we know that no long row can have more than $2h - 2$ dirty edges. Therefore, there must be at least $w - (2h - 2)$ clean edges in any long row.

As in explained in Case 2.2.1, the clean component must be continuous.

Thus, there must exist a tower of width $w - (2h - 2)$.

Since there is only one clean column, from Lemma 26 there is no tower of width 4. Hence $w - (2h - 2) < 4$.

Therefore $w < 2h+2$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+2}{2} + 2 < h+3 \leq h+2$ guards. The current strategy uses $2h$ guards and $2h \geq h+2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.2.4 *The completely clean short row contains no guards, the partly dirty short row contains two guards, one long row contains three guards and all long rows contain two guards. The partly dirty short row is adjacent to a long row containing three guards.*

The long row with three guards in this case can be used to expand the number of clean edges or dirty edges in one row; it cannot expand both. If it expands the number of dirty edges then the proof is identical to 3. If it expands the number of clean edges then consider the following: From Corollary 11 we know that no long row can have more than $2h-4$ dirty edges. Therefore, there must be at least $w-(2h-4)$ clean edges in any long row.

As in explained in Case 2.2.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h-4)$.

Since there is only one clean column, from Lemma 27 there is no tower of width 6. Hence $w-(2h-2)<6$.

Therefore $w<2h+4$. Hence we can clean the grid with no more than $w/2+2<\frac{2h+4}{2}+2<h+4\leq h+3$ guards. The current strategy uses $2h$ guards and $2h\geq h+3$ for all $h\geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.3. *At step* $t-1$*, suppose one short row is partly dirty and one is completely dirty.*

The completely dirty short row contains exactly one guard and the partly dirty short row contains exactly two guards and no long row contains more two guards, unless we have a trivial situation. By Corollary 1, no long row may contain more than $2h-2$ clean edges. R contains $w-2$ clean edges. Thus, $2h-2\geq w-2$, i.e. $w\leq 2h$.

By Lemma 7, we can clean the graph with $w/2+2\leq\frac{2h}{2}+2\leq h+2$ guards. The strategy we are considering has just been shown to use at least $2h$ guards which is greater than $h+2$ for all $h\geq 3$. So the current strategy uses at least $w/2+2$ guards.

Case 2.4. *At* $t-1$, *suppose both short rows are completely dirty.*

We consider the three non-trivial cases:

Case 2.4.1 *One short row contains one guard, the other short row contains two guards, and all long rows contain two guards.*

By Corollary 1, no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h$ guards on the graph at step t, which is greater than $h+2$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.4.2 *Both short rows contain one guard, and all long rows contain two guards.*

By Lemma 8, long row i, relative to one of the short rows, contains at most $2i+2$ clean edges.

By Lemma 8, long row j, relative to the other short row, contains at most $2j+2$ clean edges. $j = h-i-1$, hence long row j contains at most $2h-2i$ clean edges.

Let the number of clean edges in row i be c. Adding the two relations gives: $2c \leq 2h+2$, i.e. $c \leq h+1$, for any row.

But row R contains exactly $w-2$ clean edges. Hence $w-2 \leq h+1$ and $w \leq h+3$. w is even, so we have: if h is even then $w \leq h+2$, if h is odd then $w \leq h+3$.

$w/2+2$ guards are sufficient to clean the grid, i.e. $\frac{h+2}{2}+2$ if h is even, $\frac{h+3}{2}+2$ if h is odd.

But the current strategy uses at least $2h-1$ guards. Hence, for $h \geq 3$,

$2h-1 \geq w/2+2$. The current strategy uses at least $w/2+2$ guards.

Case 2.4.3 *Both short rows contain one guard, one long row contains three guards, and all other long rows contain two guards.*

By Lemma 8, long row i, relative to one of the short rows, contains at most $2i+2$ clean edges.

By Lemma 9, long row j, relative to the other short row, contains at most $2j+4$ clean edges. $j = h-i-1$, hence long row j contains at most $2h-2i+2$ clean edges.

Let the number of clean edges in row i be c. Adding the two relations gives: $2c \leq 2h+6$, i.e. $c \leq h+3$, for any row.

But row R contains exactly $w-2$ clean edges. Hence $w-2 \leq h+3$ and $w \leq h+5$. w is even, so we have: if h is even then $w \leq h+4$, if h is odd then $w \leq h+5$.

$w/2+2$ guards are sufficient to clean the grid, i.e. $\frac{h+4}{2}+2$ if h is even, $\frac{h+5}{2}+2$ if h is odd.

But the current strategy uses at least $2h$ guards. Hence, for $h \geq 3$, $2h \geq w/2+2$. The current strategy uses a least $w/2+2$ guards.

Case 2.5. *At step* $t-1$, *suppose one short row is completely dirty and one is completely clean.*

We consider the eleven non-trivial cases:

Case 2.5.1 *The completely clean short row contains no guards, the completely dirty short row contains one guard, one long row contains four guards and all other long rows contain two guards.*

By Corollary 4, since there is a completely dirty short row with one guard no long row can have more than $2h+2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h+2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h + 4$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2 + 2 \leq \frac{2h+4}{2} + 2 \leq h + 4$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h + 4$ for all $h \geq 4$, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w \leq 2h + 4 \leq 2 \cdot 3 + 4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 4 the long row cannot have more than 8 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 8/2 + 2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards, hence, the current strategy uses at least $w/2 + 2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2 + 2$ guards.

Case 2.5.2 *The completely clean short row contains no guards, the completely dirty short row contains one guard, one long row contains three* *guards and all other long rows contain two guards.*

By Corollary 2, since there is a completely dirty short row with one guard no long row can have more than $2h$ clean edges. Long row R has exactly $w - 2$ clean edges. Thus, $2h \geq w - 2$ or the configuration is not possible, i.e. $w \leq 2h + 2$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2 + 2 \leq \frac{2h+2}{2} + 2 \leq h + 3$ guards. Since there are currently $2h - 1$ guards on the graph at step t and $2h - 1 \geq h + 3$ for all $h \geq 4$, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w \leq 2h + 2 \leq 2 \cdot 3 + 2 \leq 8$. This means the graph can have a width of 8 or 6. We consider each of these widths as its own case:

Case $w = 8$

By Corollary 2 the long row can have no more than 6 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than 5 guards. Therefore, the current strategy uses at least $w/2 + 2$ guards.

Case 2.5.3 *The completely clean short row contains no guards, the completely dirty short row contains one guard, two long rows contain three guards and all other long rows contain two guards.*

By Corollary 3, since there is a completely dirty short row with one guard no long row can have more than $2h+2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h+2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h+4$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+4}{2}+2 \leq h+4$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h+4$ for all $h \geq 4$, the current strategy uses at least $w/2+2$ guards.

When $h=3$ then $w \leq 2h+4 \leq 2 \cdot 3+4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 3 the long row cannot have more than 8 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 8/2+2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards, hence, the current strategy uses at least $w/2+2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case 2.5.4 *The completely clean short row contains no guards, the completely dirty short row contains one guard and all long rows contain two guards.*

By Corollary 1, since there is a completely dirty short row with one guard no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h-2$ guards on the graph at step t and $2h-2 \geq h+2$ for all $h \geq 4$, the current strategy uses at least $w/2+2$ guards.

When $h=3$ then $w \leq 2h \leq 2 \cdot 3 \leq 6$. This means the graph can have a width of 6. By Corollary 1 the long row can have no more than 4 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case 2.5.5 *The completely clean short row contains no guards, the completely dirty short row contains two guards, one long row contains three guards and all other long rows contain two guards.*

By Corollary 6, since there is a completely dirty short row with one guard no long row can have more than $2h$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h+2$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+2}{2}+2 \leq h+3$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h+3$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.6 *The completely clean short row contains no guards, the completely dirty short row contains two guards and all long rows contain two guards.*

By Corollary 5, since there is a completely dirty short row with one guard no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h-2$ guards on the graph at step t and $2h-1 \geq h+2$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.7 *The completely clean short row contains no guards, the completely dirty short row contains three guards and all long rows contain two guards.*

By Corollary 7, since there is a completely dirty short row with one guard no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h+2$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.8 *The completely dirty short row contains one guard, the com-*

pletely clean short row contains two guards, and all long rows contain two guards.

By Corollary 1, since there is a completely dirty short row with one guard no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h+2$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.9 *The completely dirty short row contains two guards, the completely clean short row contains one guard, and all long rows contain two guards.*

By Corollary 5, since there is a completely dirty short row with two guards no long row can have more than $2h-2$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h$ guards on the graph at step t and $2h \geq h+2$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.10 *The completely dirty short row contains one guard, the completely clean short row contains one guard, and all long rows contain two guards.*

By Corollary 1, since there is a completely dirty short row with one guard no long row can have more than $2h-2$ clean edges. Long row R

has exactly $w-2$ clean edges. Thus, $2h-2 \geq w-2$ or the configuration is not possible, i.e. $w \leq 2h$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h-1$ guards on the graph at step t and $2h-1 \geq h+3$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.5.11 *The completely dirty short row contains one guard, the completely clean short row contains one guard, one long row contains three guards, and all other long rows contain two guards.*

By Corollary 2, since there is a completely dirty short row with one guard no long row can have more than $2h$ clean edges. Long row R has exactly $w-2$ clean edges. Thus, $2h \geq w-2$, i.e. $w \leq 2h+2$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+2}{2}+2 \leq h+3$ guards. Since there are $2h$ guards on the graph at step t and $2h \geq h+3$ for all $h \geq 3$, the current strategy uses at least $w/2+2$ guards.

Case 2.6. *At step $t-1$, suppose both of the short rows are completely clean.*

We consider the seventeen non-trivial cases:

Case 2.6.1 *Both short rows contain no guards, one long row contains five guards and all other long rows contain two guards.*

From Corollary 14 we know that no long row can have more than $2h+2$ dirty edges. Therefore, there must be at least $w-(2h+2)$ clean edges in any long row.

Furthermore, from Lemma 22 we know that any long row j where $1 \leq j < i$ has less dirty edges than in long row i and, from that proof we know that in order to achieve the maximum number of clean edges in each long row, a "pyramid" of clean edges is required. Finally, because there 2 guards per long row for all long rows except one, the clean component must be continuous within each long row.

Thus, there must exist a tower of width $w - (2h + 2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h + 2) < 4$.

Therefore $w < 2h + 6$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+6}{2} + 2 < h + 5 \leq h + 4$ guards. The current strategy uses $2h$ guards and $2h \geq h + 4$ for all $h \geq 4$, thus, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w < 2h + 6 < 2 \cdot 3 + 6 < 12$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur. Thus, this case is not possible.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 8/2 + 2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards, hence, the strategy uses at least $w/2 + 2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the strategy uses at least $w/2+2$ guards.

Case 2.6.2 *Both short rows contain no guards, one long row contains four guards and all other long rows contain two guards.*

From Corollary 13 we know that no long row can have more than $2h$ dirty edges. Therefore, there must be at least $w-2h$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-2h$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-2h<4$.

Therefore $w<2h+4$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+4}{2}+2 < h+4 \leq h+3$ guards. The current strategy uses $2h-1$ guards and $2h-1 \geq h+3$ for all $h \geq 4$, thus, the current strategy uses at least $w/2+2$ guards.

When $h=3$ then $w<2h+4<2\cdot 3+4<10$. This means the graph can have a width of 8 or 6. We consider each of these widths as its own case:

Case $w=8$

From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur.

Thus, this case is not possible.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than $2h - 1 = 2 \cdot 3 - 1 = 5$ guards. Therefore, the strategy uses at least $w/2 + 2$ guards.

Case 2.6.3 *Both short rows contain no guards, one long row contains three guards and all other long rows contain two guards.*

From Corollary 12 we know that no long row can have more than $2h - 2$ dirty edges. Therefore, there must be at least $w - (2h - 2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h - 2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h - 2) < 4$.

Therefore $w < 2h+2$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+2}{2} + 2 < h + 3 \leq h + 2$ guards. The current strategy uses $2h - 2$ guards and $2h - 2 \geq h + 2$ for all $h \geq 4$, thus, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w < 2h+2 < 2 \cdot 3+2 < 8$. This means the graph can have a width of 6. From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur. Thus, this case is not possible.

Case 2.6.4 *Both short rows contain no guards, and all other long rows contain two guards.*

From Corollary 11 we know that no long row can have more than $2h - 4$ dirty edges. Therefore, there must be at least $w - (2h - 4)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h - 4)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h - 4) < 4$.

Therefore $w < 2h$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h}{2} + 2 < h + 2 \leq h + 1$ guards. The current strategy uses $2h - 3$ guards and $2h - 3 \geq h + 1$ for all $h \geq 4$, thus, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w < 2h < 2 \cdot 3 < 6$. However, walls are undefined for widths less than 6 and thus this situation is not possible.

Case 2.6.5 *Both short rows contain no guards, one long row contains three guards, one long row contains four guards and all other long rows contain two guards.*

From Corollary 15 we know that no long row can have more than $2h+2$ dirty edges. Therefore, there must be at least $w-(2h+2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h + 2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h + 2) < 4$.

Therefore $w < 2h+6$. Hence we can clean the grid with no more than

$w/2+2 < \frac{2h+6}{2}+2 < h+5 \leq h+4$ guards. The current strategy uses $2h$ guards and $2h \geq h+4$ for all $h \geq 4$, thus, the current strategy uses at least $w/2+2$ guards.

When $h = 3$ then $w < 2h+6 < 2 \cdot 3+6 < 12$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur. Thus, this case is not possible.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 8/2+2 = 6$. The current strategy uses no less than $2h = 2{\cdot}3 = 6$ guards, hence, the strategy uses at least $w/2+2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the strategy uses at least $w/2+2$ guards.

Case 2.6.6 *Both short rows contain no guards, two long rows contain three guards and all other long rows contain two guards.*

From Corollary 16 we know that no long row can have more than $2h$

dirty edges. Therefore, there must be at least $w - 2h$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - 2h$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - 2h < 4$.

Therefore $w < 2h+4$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+4}{2} + 2 < h + 4 \leq h + 3$ guards. The current strategy uses $2h - 1$ guards and $2h - 1 \geq h + 3$ for all $h \geq 4$, thus, the current strategy uses at least $w/2 + 2$ guards.

When $h = 3$ then $w < 2h + 4 < 2 \cdot 3 + 4 < 10$. This means the graph can have a width of 8 or 6. We consider each of these widths as its own case:

Case $w = 8$

From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur. Thus, this case is not possible.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than $2h - 1 = 2 \cdot 3 - 1 = 5$ guards, hence, the strategy uses at least $w/2 + 2$ guards.

Case 2.6.7 *Both short rows contain no guards, three long rows contain* *three guards and all other long rows contain two guards.*

From Corollary 17 we know that no long row can have more than $2h+2$ dirty edges. Therefore, there must be at least $w-(2h+2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h+2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-(2h+2)<4$.

Therefore $w<2h+6$. Hence we can clean the grid with no more than $w/2+2<\frac{2h+6}{2}+2<h+5\leq h+4$ guards. The current strategy uses $2h$ guards and $2h\geq h+4$ for all $h\geq 4$, thus, the current strategy uses at least $w/2+2$ guards.

When $h=3$ then $w<2h+6<2\cdot 3+6<12$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w=10$

From Lemma 28 if a tower of width 4 exists then there must be two clean columns. Since there is only one long row, that means that this long row cannot have more than 3 clean edges. But since the long row is adjacent to the completely clean row and the completely clean row has no guards recontamination must occur. Thus, this case is not possible.

Case $w=8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2=8/2+2=6$. The current strategy uses no less than $2h=2\cdot 3=6$ guards, hence, the strategy uses at least $w/2+2$ guards.

Case $w=6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2=6/2+2=5$. The current strategy uses no less than 6 guards. Therefore, the strategy uses at least $w/2+2$ guards.

Case 2.6.8 *One short row contains no guards, the other short row contains one guard, one long row contains four guards and all other long rows contain two guards.*

From Corollary 13 we know that no long row can have more than $2h$ dirty edges. Therefore, there must be at least $w-2h$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-2h$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-2h<4$.

Therefore $w<2h+4$. Hence we can clean the grid with no more than $w/2+2<\frac{2h+4}{2}+2<h+4\leq h+3$ guards. The current strategy uses $2h$ guards and $2h\geq h+3$ for all $h\geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.6.9 *One short row contains no guards, the other short row contains one guard, one long row contains three guards and all other long rows contain two guards.*

From Corollary 12 we know that no long row can have more than $2h-2$ dirty edges. Therefore, there must be at least $w-(2h-2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h - 2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h - 2) < 4$.

Therefore $w < 2h + 2$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+2}{2} + 2 < h + 3 \leq h + 2$ guards. The current strategy uses $2h - 1$ guards and $2h - 1 \geq h + 2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.6.10 *One short row contains no guards, the other short row contains one guard and all other long rows contain two guards.*

From Corollary 11 we know that no long row can have more than $2h - 4$ dirty edges. Therefore, there must be at least $w - (2h - 4)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h - 4)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h - 4) < 4$.

Therefore $w < 2h$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h}{2} + 2 < h + 2 \leq h + 1$ guards. The current strategy uses $2h - 2$ guards and $2h - 2 \geq h + 1$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.6.11 *One short row contains no guards, the other short row contains one guard, two long rows contain three guards and all other long rows contain two guards.*

From Corollary 16 we know that no long row can have more than $2h$ dirty edges. Therefore, there must be at least $w - 2h$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - 2h$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - 2h < 4$.

Therefore $w < 2h + 4$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+4}{2} + 2 < h + 4 \leq h + 3$ guards. The current strategy uses $2h$ guards and $2h \geq h + 3$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.6.12 *One short row contains no guards, the other short row contains two guards and one long row contains three guards and all other long rows contain two guards.*

From Corollary 12 we know that no long row can have more than $2h - 2$ dirty edges. Therefore, there must be at least $w - (2h - 2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - (2h - 2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - (2h - 2) < 4$.

Therefore $w < 2h + 2$. Hence we can clean the grid with no more than $w/2 + 2 < \frac{2h+2}{2} + 2 < h + 3 \leq h + 2$ guards. The current strategy uses $2h$ guards and $2h \geq h + 2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2 + 2$ guards.

Case 2.6.13 *One short row contains no guards, the other short row contains two guards and all long rows contain two guards.*

From Corollary 11 we know that no long row can have more than

$2h-4$ dirty edges. Therefore, there must be at least $w-(2h-4)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h-4)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-(2h-4)<4$.

Therefore $w<2h$. Hence we can clean the grid with no more than $w/2+2<\frac{2h}{2}+2<h+2\leq h+1$ guards. The current strategy uses $2h-1$ guards and $2h-1\geq h+1$ for all $h\geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.6.14 *One short row contains no guards, the other short row contains three guards and all long rows contain two guards.*

From Corollary 11 we know that no long row can have more than $2h-4$ dirty edges. Therefore, there must be at least $w-(2h-4)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h-4)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-(2h-4)<4$.

Therefore $w<2h$. Hence we can clean the grid with no more than $w/2+2<\frac{2h}{2}+2<h+2\leq h+1$ guards. The current strategy uses $2h$ guards and $2h\geq h+1$ for all $h\geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.6.15 *One short row contains one guard, the other short row contains two guards, and all long rows contain two guards.*

From Corollary 8 we know that no long row can have more than $2h-2$

dirty edges. Therefore, there must be at least $w-(2h-2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h-2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-(2h-2) < 4$.

Therefore $w < 2h+2$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+2}{2}+2 < h+3 \leq h+2$ guards. The current strategy uses $2h$ guards and $2h \geq h+2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.6.16 *Both short rows contain one guard, and all long rows contain two guards.*

From Corollary 8 we know that no long row can have more than $2h-2$ dirty edges. Therefore, there must be at least $w-(2h-2)$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w-(2h-2)$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w-(2h-2) < 4$.

Therefore $w < 2h+2$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+2}{2}+2 < h+3 \leq h+2$ guards. The current strategy uses $2h-1$ guards and $2h-1 \geq h+2$ for all $h \geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 2.6.17 *Both short rows contain one guard, one long row contains three guards, and all other long rows contain two guards.*

From Corollary 9 we know that no long row can have more than $2h$

dirty edges. Therefore, there must be at least $w - 2h$ clean edges in any long row.

As explained in Case 2.6.1, the clean component must be continuous. Thus, there must exist a tower of width $w - 2h$.

Since there is only one clean column, from Lemma 28 there is no tower of width 4. Hence $w - 2h < 4$.

Therefore $w < 2h+4$. Hence we can clean the grid with no more than $w/2+2 < \frac{2h+4}{2}+2 < h+4 \leq h+3$ guards. The current strategy uses $2h$ guards and $2h \geq h+3$ for all $h \geq 3$, thus, the current strategy uses at least $w/2+2$ guards.

Case 3 *Suppose C^t contains a long row R and two columns.*

At step $t-1$, R is partly dirty and there exists exactly one completely clean column.

Case 2 is divided into two subcases: either a clean column exists at $t-1$ or not. In the subcase of a clean column existing, the situation at step $t-1$ in Case 2 is that one long row is almost clean and all long rows are partly dirty and there exists one clean column; this is identical to the current situation in this case at $t-1$. Hence, even though the two cases differ at t, since the proof is performed at $t-1$ the proofs for both of these cases is identical.

Case 4 *Suppose at step t, two columns are clean but no long row is clean.*

At $t-1$, one column is clean and no long rows are clean. Let R be the row where the guard is placed to clean the column at step t. R can be either a long row or a short row.

We consider six possible cases depending on whether the two short rows are partly dirty, completely clean or completely dirty at step $t-1$.

Case 4.1. *At step $t-1$, suppose both short rows are partly dirty.*

Then each short row contains at least two guards, since the rows wrap-around and contain both clean and dirty edges. Hence, there are at least $2h+1$ guards on the graph at step t.

Case 4.2. *At step $t-1$, suppose one short row is partly dirty and one is completely clean.*

The proof of Case 2 makes uses of the following facts: there exists a partly dirty short row with exactly two guards, there exists a completely clean short row with zero or one guard, and every long row is partly dirty and contains at least two guards. These conditions haven't changed and thus it follows that the current strategy uses at least $w/2+2$ guards.

Case 4.3. *At step $t-1$, suppose one short row is partly dirty and one is completely dirty.*

The partly dirty row must contain at least two guards.

Because step t is a placement, the completely dirty row will either remain completely dirty or, if the placement is on the completely dirty short row and adjacent to the existing guard on that row, it will become partly dirty. If it is partly dirty at step t then it has two guards. If it is completely dirty at t then it has two guards since there are two clean columns at step t.

Thus, at step t, the graph contains at least $2h$ guards.

All the guards on a partly dirty row are critical, so all the guards on one of the short rows and all of the long rows are critical. If the other short row is partly dirty then the guards are critical as well. If the other short is completely dirty, then both guards on it may be critical or are incident with only dirty edges. If a guard is incident with only dirty edges it will not be removed since it would need to be put back again later and we only consider progressive strategies. Thus, at least $2h$ guards will not be removed.

If $t+1$ is a removal, there must be $2h+1$ guards on the graph at step t, since at least $2h$ guards will not be removed. If $t+1$ is a placement, then there are at least $2h+1$ guards on the graph at step $t+1$.

Case 4.4. *At* $t-1$, *suppose both short rows are completely dirty.*

If the guard placed at step t is placed in a short row adjacent to the guard in that short row then one of the short rows will become partly dirty. Otherwise, both rows remain completely dirty at step t.

Each completely dirty short row at t must contain two guards at step t because each has two vertices in common with the clean columns.

If there is a partly dirty short row at step t, it must contain two guards. Thus, there are at least $2h$ guards in total at step t.

All the guards on a partly dirty row are critical, so all the guards in long rows are critical. If the one short row is partly dirty then the guards are critical as well. On any completely dirty short row, both guards must be incident to clean edges in the columns and hence are critical.

If $t+1$ is a removal, there must be $2h+1$ guards on the graph at step t, since at least $2h$ guards are critical. If $t+1$ is a placement, then there are at least $2h+1$ guards on the graph at step $t+1$.

Case 4.5. *At step* $t-1$, *suppose one short row is completely dirty and one is completely clean.*

We consider the eleven non-trivial cases:

Case 4.5.1 *The completely clean short row contains no guards, the com-*

pletely dirty short row contains one guard, one long row contains four guards and all other long rows contain two guards.

The long row with four guards can either be used to increase the number of clean edges or the number of dirty edges, but it cannot do both. Since both cases produce the same result, let the long row with four guards be used to increase the number of dirty edges.

By Lemma 21, a long row i distance from the completely clean short row can have no more than $2i+4$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+4+2j+2 \leq 2i+4+2(h-i-1)+2 \leq 2h+4$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+4}{2}+2 \leq h+4$ guards. Since there are currently $2h$ guards on the graph at step t, and $2h \geq h+4$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph.

When $h = 3$ then $w \leq 2h+4 \leq 2 \cdot 3+4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 11 the long row cannot have more than 8 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 8/2 + 2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards. Therefore, the current strategy uses at least $w/2 + 2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2 + 2$ guards.

Case 4.5.2 *The completely clean short row contains no guards, the completely dirty short row contains one guard, one long row contains three guards and all other long rows contain two guards.*

The long row with three guards can either be used to increase the number of clean edges or the number of dirty edges, but it cannot do both. Since both cases produce the same result, let the long row with three guards be used to increase the number of dirty edges.

By Lemma 20, a long row i distance from the completely clean short row can have no more than $2i+2$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2+2j+2 \leq 2i+2+2(h-i-1)+2 \leq 2h+2$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2 + 2 \leq \frac{2h+2}{2} + 2 \leq h + 3$ guards. Since there are currently $2h - 1$ guards on the graph at step t, and $2h - 1 \geq h + 3$ for all $h \geq 4$ there must be at least $w/2 + 2$ guards on the graph.

When $h = 3$ then $w \leq 2h + 2 \leq 2 \cdot 3 + 2 \leq 8$. This means the graph can have a width of 8 or 6. We consider each of these widths as its own case:

Case $w = 8$

By Lemma 9 the long row cannot have more than 6 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than $2h - 1 = 2 \cdot 3 - 1 = 5$ guards. Therefore, the current strategy uses at least $w/2 + 2$ guards.

Case 4.5.3 *The completely clean short row contains no guards, the completely dirty short row contains one guard, two long rows contain three guards and all other long rows contain two guards.*

The two long rows with three guards can either be used to increase the number of clean edges or the number of dirty edges, or one can increase the number of clean edges and the other the number of dirty edges. Regardless, since all cases produce the same result, let them be used to increase the number of dirty edges.

By Lemma 24, a long row i distance from the completely clean short row can have no more than $2i+4$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j + 2$ clean edges. But $j = h - i - 1$. Since w is the sum of the clean and dirty edges, $w \leq 2i+4+2j+2 \leq 2i+4+2(h-i-1)+2 \leq 2h+4$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+4}{2}+2 \leq h+4$ guards. Since there are currently $2h$ guards on the graph at step t, and $2h \geq h+4$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph.

When $h=3$ then $w \leq 2h+4 \leq 2\cdot 3+4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

Since there is only one long row there can only be one long row with three guards. By Lemma 9 the long row cannot have more than 6 clean edges. This means that at least one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 8/2+2 = 6$. The current strategy uses no less than $2h = 2\cdot 3 = 6$ guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case 4.5.4 *The completely clean short row contains no guards, the com-*

pletely dirty short row contains one guard and all long rows contain two guards.

By Lemma 19, a long row i distance from the completely clean short row can have no more than $2i$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2j+2 \leq 2i+2(h-i-1)+2 \leq 2h$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h-2$ guards on the graph at step t, and $2h-2 \geq h+2$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph. When $h = 3$ then $w \leq 2h \leq 2 \cdot 3 \leq 6$. By Lemma 8 the long row cannot have more than 4 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case 4.5.5 *The completely clean short row contains no guards, the completely dirty short row contains two guards, one long row contains three guards and all other long rows contain two guards.*

The long row with three guards can either be used to increase the number of clean edges or the number of dirty edges, but it cannot do both. We will use it to increase the number of clean edges since that is the worse case.

By Lemma 19, a long row i distance from the completely clean short row can have no more than $2i$ dirty edges. By Lemma 14, a long row j distance from the completely dirty short row can have no more than

$2j+6$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2j+6 \leq 2i+2(h-i-1)+6 \leq 2h+4$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+4}{2}+2 \leq h+4$ guards. Since there are currently $2h$ guards on the graph at step t, and $2h \geq h+4$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph.

When $h = 3$ then $w \leq 2h+4 \leq 2 \cdot 3+4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 14 the long row cannot have more than 8 clean edges. This means that one of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 8/2+2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards. Therefore, the current strategy uses least $w/2+2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case 4.5.6 *The completely clean short row contains no guards, the com-*

pletely dirty short row contains two guards and all long rows contain two guards.

By Lemma 19, a long row i distance from the completely clean short row can have no more than $2i$ dirty edges. By Lemma 13, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2j+2 \leq 2i+2(h-i-1)+2 \leq 2h$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h-1$ guards on the graph at step t, and $2h-1 \geq h+2$ for all $h \geq 3$ there must be at least $w/2+2$ guards on the graph.

Case 4.5.7 *The completely clean short row contains no guards, the completely dirty short row contains three guards and all long rows contain two guards.*

By Lemma 19, a long row i distance from the completely clean short row can have no more than $2i$ dirty edges. By Lemma 15, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2j+2 \leq 2i+2(h-i-1)+2 \leq 2h$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h}{2}+2 \leq h+2$ guards. Since there are currently $2h$ guards on the graph at step t, and $2h \geq h+2$ for all $h \geq 3$ there must be at least $w/2+2$ guards on the graph.

Case 4.5.8 *The completely dirty short row contains one guard, the completely clean short row contains two guards, and all long rows contain two guards.*

By Lemma 8, a long row i distance from the completely clean short row can have no more than $2i + 2$ clean edges. By Lemma 18, a long row j distance from the completely dirty short row can have no more than $2j+4$ dirty edges. But $j = h-i-1$. Since w is the sum of the number of clean and dirty edges, $w \leq 2i+2+2j+4 \leq 2i+2+2(h-i-1)+4 \leq 2h + 4$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2 + 2 \leq \frac{2h+4}{2} + 2 \leq h + 4$ guards. There are $2h$ guards on the graph at step t. $2h \geq h + 4 \geq w/2 + 2$ for all $h \geq 4$. When $h = 3$ then $w \leq 2h + 4 \leq 2 \cdot 3 + 4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 8 the long row cannot have more than 4 clean edges. This means that three of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has no guards and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 8/2 + 2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards, hence, the current strategy uses at least $w/2 + 2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case 4.5.9 *The completely dirty short row contains two guards, the completely clean short row contains one guard, and all long rows contain two guards.*

By Corollary 8, since there is a completely clean short row with one guard no long row can have more than $2h-2$ dirty edges. The long row next to the completely dirty short row must have at least $w-4$ dirty edges by Lemma 12 and Lemma 8. Thus, $2h-2 \geq w-4$ or recontamination will occur, i.e. $w \leq 2h+2$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+2}{2}+2 \leq h+3$ guards. Since there are currently $2h$ guards on the graph at step t, for all $h \geq 3$ there must be at least $w/2+2$ guards on the graph.

Case 4.5.10 *The completely dirty short row contains one guard, the completely clean short row contains one guard, and all long rows contain two guards.*

By Lemma 16, a long row i distance from the completely clean short row can have no more than $2i+2$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j+2$ clean edges. But $j = h-i-1$. Since w is the sum of the clean and dirty edges, $w \leq 2i+2+2j+2 \leq 2i+2+2(h-i-1)+2 \leq 2h+2$. We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+2}{2}+2 \leq h+3$ guards. Since there are currently $2h-1$ guards on the graph at step t, and $2h-1 \leq h+3$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph.

When $h = 3$ then $w \leq 2h + 2 \leq 2 \cdot 3 + 2 \leq 8$. This means the graph can have a width of 8, or 6. We consider each of these widths as its own case:

Case $w = 8$

By Lemma 8 the long row cannot have more than 4 clean edges. This means that two of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has only one guard and hence recontamination must occur.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2 + 2 = 6/2 + 2 = 5$. The current strategy uses no less than $2h - 1 = 2 \cdot 3 - 1 = 5$ guards. Hence, the current strategy uses at least $w/2 + 2$ guards.

Case 4.5.11 *The completely dirty short row contains one guard, the completely clean short row contains one guard, one long row contains three guards, and all other long rows contain two guards.*

The long row with three guards can either be used to increase the number of clean edges or the number of dirty edges, but it cannot do both. Since both cases produce the same result, let the long row with three guards be used to increase the number of dirty edges.

By Lemma 17, a long row i distance from the completely clean short row can have no more than $2i+4$ dirty edges. By Lemma 8, a long row j distance from the completely dirty short row can have no more than $2j + 2$ clean edges. But $j = h - i - 1$. Since w is the sum of the clean and dirty edges, $w \leq 2i+4+2j+2 \leq 2i+4+2(h-i-1)+2 \leq 2h+4$.

We know from Lemma 7 that an optimal strategy exists that cleans the graph using only $w/2+2 \leq \frac{2h+4}{2}+2 \leq h+4$ guards. Since there are currently $2h$ guards on the graph at step t, and $2h \geq h+4$ for all $h \geq 4$ there must be at least $w/2+2$ guards on the graph.

When $h=3$ then $w \leq 2h+4 \leq 2 \cdot 3+4 \leq 10$. This means the graph can have a width of 10, 8, or 6. We consider each of these widths as its own case:

Case $w = 10$

By Lemma 9 the long row cannot have more than 6 clean edges. This means that two of the connecting edges between the completely clean short row can the long row must be dirty. But the completely clean short row has only one guard and hence recontamination must occur.

Case $w = 8$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 8/2+2 = 6$. The current strategy uses no less than $2h = 2 \cdot 3 = 6$ guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case $w = 6$

By Lemma 7 we know that the number of guards needed to clean this graph is no more than $w/2+2 = 6/2+2 = 5$. The current strategy uses no less than 6 guards. Therefore, the current strategy uses at least $w/2+2$ guards.

Case 4.6. *At step $t-1$, suppose both of the short rows are completely clean.*

The proof of Case 6 makes uses of the following facts: there exists two completely clean short rows and every long row is partly dirty and contains at least two guards. These conditions haven't changed and thus it follows that the current strategy uses at least $w/2+2$ guards. □

Theorem 6. *Let G be a wall,* $\text{ns}(G) = \min(2h+1, w/2+2)$, *where $h, w \geq 4$.*

Proof. From Lemma 7 we get an upper bound $\text{ns}(G) \leq \min(h+1, 2w+1)$ and from Lemma 29 we get a lower bound $\text{ns}(G) \geq \min(h+1, 2w+1)$, therefore, $\text{ns}(G) = \min(h+1, 2w+1)$. □

3.5 Node Search Number on Tori

We outline the obvious strategy for cleaning an (h, w)-torus and go on to show that this strategy is optimal, so long as $h \neq w$.

Lemma 30. *If G is an (h,w)-torus then $ns(G) \leq \min(2h+1, 2w+1)$.*

Proof. Consider filling two adjacent columns with guards and "sweeping" one guard column around the width of the torus until it comes up against the other side of the stationary guarded column. The movement of one column of guards to the adjacent dirty column is accomplished by the use of one extra guard. The placement of the extra guard on a node in the adjacent dirty column frees one guard in the clean column which can be used on the next node in the adjacent dirty column, and so on till the adjacent column is completely clean and completely occupied by guards. Exactly $2h+1$ guards are used and since the torus is symmetrical with respect to rows and columns the same process can sweep a row around the height of the torus, using $2w+1$ guards. □

We continue by showing that this strategy is optimal for non-equidimensional tori.

Lemma 31. *If G is an (h,w)-torus where $h \neq w$ then $ns(G) \geq \min(2h+1, 2w+1)$.*

Proof. Suppose $h < w$ and let step t be the first step at which the set of clean edges defines a component C^t which includes two columns or (inclusive) two rows. We consider two cases, either C^t contains two columns and up to two rows or it contains no more than one column and two rows.

- Consider the case where C^t contains two completely clean rows and no more than one completely clean column. Then there are at least $w - 1$ columns that are either partly or completely dirty.

  Any partly dirty column must contain at least two critical guards. Any completely dirty column also contains two vertices incident with both clean edges (in the two completely clean rows) and dirty edges (in the dirty column itself). Hence every completely dirty column contains at least two critical guards. Hence there are at at least $2(w - 1)$ critical guards on the graph at step t.

  Since $w > h$ there must be at least $2h$ critical guards on the graph at step t. If there are more than $2h$ guards we are done. If there are exactly $2h$ guards then, by the criticality principle, the next move must be a placement. So $2h+1$ guards are on the graph at step $t + 1$.

- Suppose C^t contains two columns, say H and H' and no more than two rows. At the completion of step $t - 1$ there is only one completely clean column, say H, one almost clean column, H', and no more than one completely clean row.

  Let R be the row where a guard is placed at step t to clean H'. It must be that, at step $t - 1$, H' contains exactly two dirty edges incident with a node v which receives a guard at step t. Further, R cannot be completely clean since v would then be incident with both clean and dirty edges at step $t - 1$ and be unguarded. We consider two possibilities, either there is a completely clean row at step $t - 1$ or not.

– Consider the case where at step $t-1$ one row, say R', is completely clean. R' cannot be the same row as R because R is not completely clean at $t-1$. All rows other than R and R' are either partly or completely dirty. Any partly dirty row must contain at least two guards. Any completely dirty row also contains two vertices incident to both clean edges (in the completely clean and almost clean columns) and dirty edges (in the dirty row itself). Hence every completely dirty row contains at least two guards. Hence all rows other than R and R' must contain at least two guards at step $t-1$.

  R' is the only completely clean row, hence it is adjacent to a row containing at least two adjacent dirty edges, incident with an unguarded vertex. Hence R' must contain at least one guard, adjacent to that unguarded vertex. R is completely or partly dirty, and hence must contain at least one guard because R intersects the completely clean column. Hence R and R' each contain at least one guard and there are at at least $2h-2$ guards on the graph at step $t-1$.

  If R contains three or more guards then there are at least $2h+1$ guards on the graph at step t, and we are done. So we need consider two cases, R contains one or two guards.

  * If R contains just one guard, then R is completely dirty. We proceed by showing that then there cannot exist a completely clean row R', contradicting our initial assumptions. Consider the rows adjacent to R. If every row contains just two guards, then there can be no more than two clean edges in these rows (between the two guards), because all the edges in R are dirty. Likewise, there can be no more than $2i$ clean edges in rows whose shortest distance from R is i.

One row could contain three guards, yet leaving no more than $2h - 1$ guards on the graph at step $t - 1$. This extra guard can increase the number of clean edges from $2i$ to $2i + 1$. However, there are two directions to go from R to R' and the row with three guards can only aid in one of those directions. We will choose the direction that does not contain the row with three guards. Thus, there can only be $2i$ clean edges.

Then there can be no more than $2i$ clean edges in rows whose shortest distance from R is i. If h is even, the maximum shortest distance from R to R' is $h/2$. So the maximum number of clean edges in row R' no more than $2h/2 = h$. But R' contains w clean edges and $w > h$. So this configuration of guards and clean edges is not possible.

If h is odd, the maximum shortest distances from R to R' are $(h-1)/2$ for the longer route and $(h-3)/2$ for the shorter route. In the worst case scenario, the row with three guards is in the shorter route. Thus, at the end of the long route there can be $2(h-1)/2 = h-1$ clean edges and at the end of the short route there an be $2(h-3)/2+1 = h-2$ clean edges. $h - 2 < h - 1 < w$, so this configuartion of guards and clean edges is not possible.

* Finally, suppose R contains two guards so that R' contains just one guard. We proceed by showing that there cannot exist sufficient dirty edges to prevent the existence of more than two columns, contradicting our assumptions for this case. Because R' is completely clean and contains just one guard and adjacent rows contain just two guards, the adjacent rows can contain to more than two dirty edges. Likewise, rows distant i from R' can contain no more than $2i$ dirty edges.

If h is odd, the maximum shortest distance between R' and any other row is $(h-1)/2$ and so the maximum number of dirty edges in any row is $h-1$. If h is even, the maximum shortest distance between R' and any other row is $h/2$ and so the maximum number of dirty edges in any row is h. Hence the set of dirty edges is necessarily a subset of a pair of "pyramids" as illustrated in Figure 3.35. The base of the pyramid contains no more than h dirty edges and $h < w$. Consequently there are at least two completely clean columns at step $t-1$, contradicting our initial assumptions. Hence this configuration of guards and clean edges is not possible.

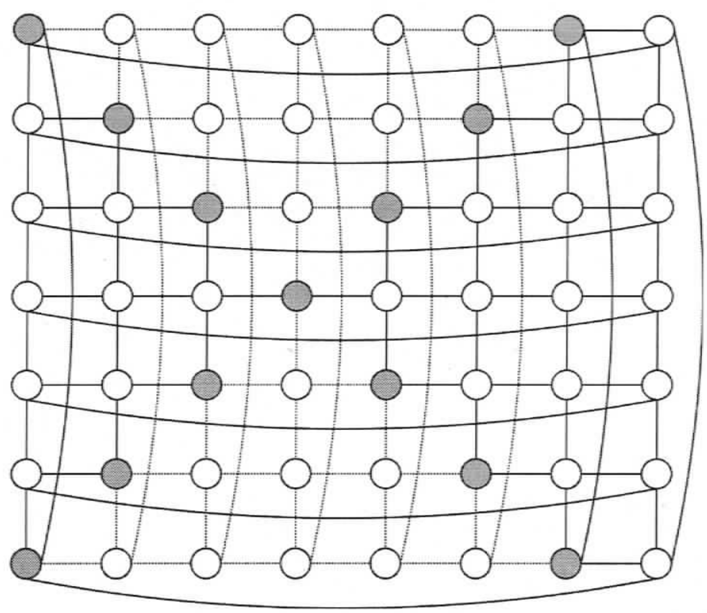


Figure 3.35: There must exist at least three clean columns

- Suppose there are no completely clean rows at step $t-1$. By the existing reasoning, every row other than R must contain at least two critical guards. Hence there are at at least $2h-1$ critical guards on the graph at step $t-1$. If R contains two guards at step $t-1$, there are at least $2h$ guards at step t and we are done. So suppose there is just one guard on R and no more than two on all other rows. Since R contains just one guard and is not completely clean, it must be completely dirty. So the placement to v, which is in H' and in R, must leave both v and the node with other guard on R adjacent to a dirty edge, i.e., there are two critical guards on R at step t. At step $t-1$ there are two critical guards on H', both adjacent to v. Because the rows containing these guards contain no more than two guards both of these guards are incident with a dirty row edge. The guard placement to v cannot clean either of these row edges. Hence these two guards remain critical at step t.

By the criticality principle, all $t-1$ guards on the graph at step $t-1$ are critical. We have just shown that the guard added at step t is critical and that the guards adjacent to v remain critical. Hence the $2h$ guards on the graph at step t are all critical. Hence, by the criticality principle, the next move is another placement.

Because the torus is symmetrical with respect to rows and columns, if $w < h$ the above proof, with rows and columns exchanged, will show that at least $2w+1$ guards are necessary. □

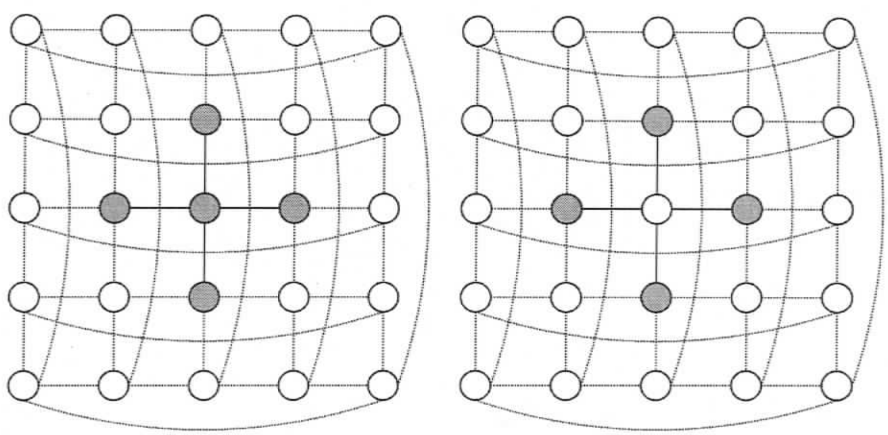
We have found that the obvious cleaning strategy for tori is not optimal for equi-dimensional, (h, h)-tori. We describe an unobvious strategy that is superior.

Lemma 32. *If G is an (h, h)-torus then $sn(G) \leq 2h$.*

Proof. We describe a searching strategy that uses $2h$ guards. The strategy comprises three parts: the initialization, the *expansion of the diamond* and the *cleaning of the corners.* By a *diamond* we mean a clean component defined by four equal length, diagonal sides. We refer to the number of nodes in a side as the *dimension* of the diamond. A diamond of dimension d contains $4(d-1)$ nodes which are incident with both clean and dirty edges (the so called *exposed* nodes), and which must therefore each be occupied by a guard. See Figure 3.37(a) and 3.37(f). We first consider the case where h is odd.

- *Initialization*

  We create a diamond of dimension two by placing a guard on some node and its four neighbours, at which point the first guard can be removed. The cost is five guards, which is less than $2h$ for all $h \geq 3$, i.e., for all (h, h)-tori. See Figure 3.36.



(a) (b)

Figure 3.36: Initializing the search

- *Expansion of the diamond*

  We observe that, so long as $2d - 1 < h$, a diamond of dimension d can be expanded to a diamond of dimension $d+1$ using just six guards over and above those $4(d-1)$ guards already on the exposed nodes of the smaller diamond, This is done as follows.

  – Place three new guards on the neighbours of the uppermost apex of the diamond so as to clean the three dirty edges incident with that apex node. Remove the guard from the apex.

- Place the released guard so as to clean the one dirty edge incident with the uppermost node in the top right side of the old diamond, thus releasing another guard. Continue expanding the top ride side of the diamond in this manner until the rightmost apex is reached.
- Use one more new guard and the just released guard to clean the two dirty edges incident with the rightmost apex node.
- Expand the bottom right side in the same way as the top right side was expanded, until the bottom apex is reached.
- Use one more new guard and the just released guard to clean the two dirty edges incident with the bottom apex node.
- Expand the remaining two sides and clean the edges incident with the leftmost apex node in the same way.

The entire expansion process is illustrated in Figure 3.37. Note that we use three extra guards at the start of this process plus one extra for each to the remaining apex nodes, i.e., six extra guards in total.

The diamond is repeatedly expanded until it encompasses an entire row and an entire column. At that point $d = \lceil h/2 \rceil$ and hence the dimension of the previous diamond was $d = \lceil h/2 \rceil - 1$. Consequently the maximum number of guards used during the last expansion was $4(\lceil h/2 \rceil - 1 - 1) + 6$, i.e., $2h$, since h is odd.

We also note that that there are $4(\lceil h/2 \rceil - 1)$ guards on the grid at the completion of the process, i.e., $2h - 2$ guards, since h is odd.

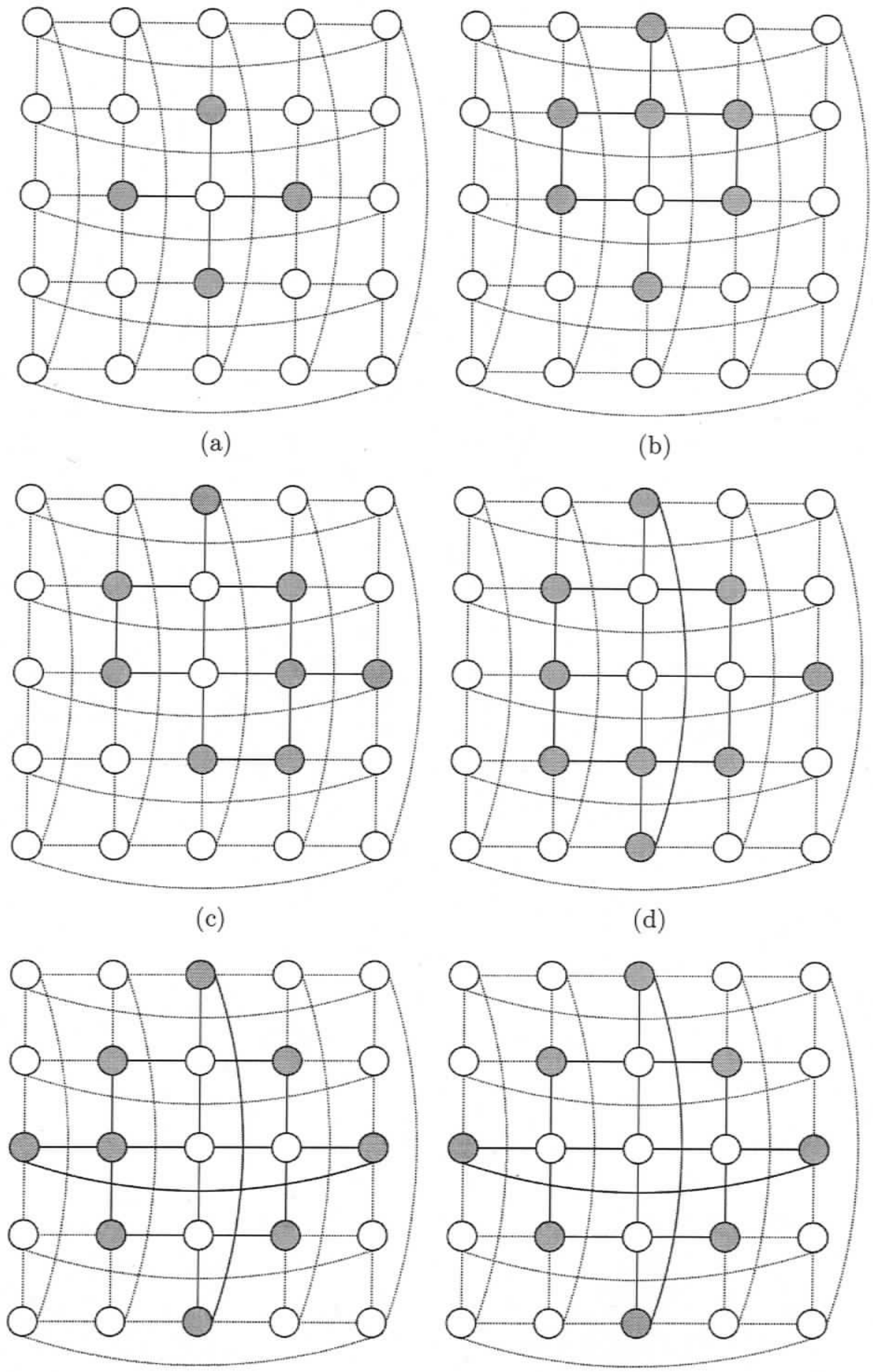


Figure 3.37: Expanding a diamond

- *Cleaning the corners*

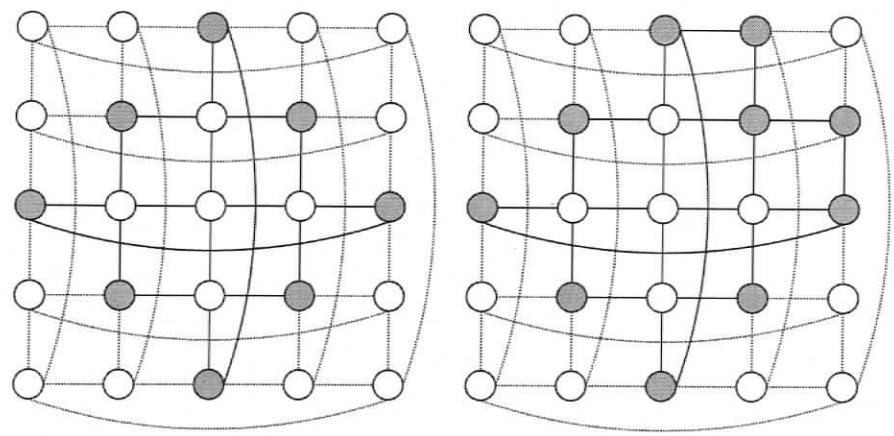
  We clean the remaining dirty edges using just two extra guards over and above those on the grid at the end of the previous phase, i.e., using $2h$ guards overall. At the end of the expansion phase, there are $2h-2$ guarded nodes, each incident with two dirty edges. Consider the uppermost guarded node in the top right side of the diamond and next to the apex. Place two extra guards so as to clean the two dirty edges incident with this node. This releases one guard, which can be used to release the guard next in line in the side, because the next guarded node is now incident with just one dirty edge. Note that this is also true for the apex nodes because these are now also incident with just one dirty edge. Hence, this process can be repeated around the entire periphery of the diamond, leaving four distinct sides, each with $\lfloor h/2 \rfloor$ nodes, i.e., $2h-2$ nodes in total. The process is illustrated in Figure 3.38.

  The use of two extra guards, in the same manner, can now reduce the number of nodes in each side by one, at which point there are only $2h-6$ occupied nodes. Again only two extra guards are necessary for the next reduction, and so on till all the remaining dirty edges are clean.

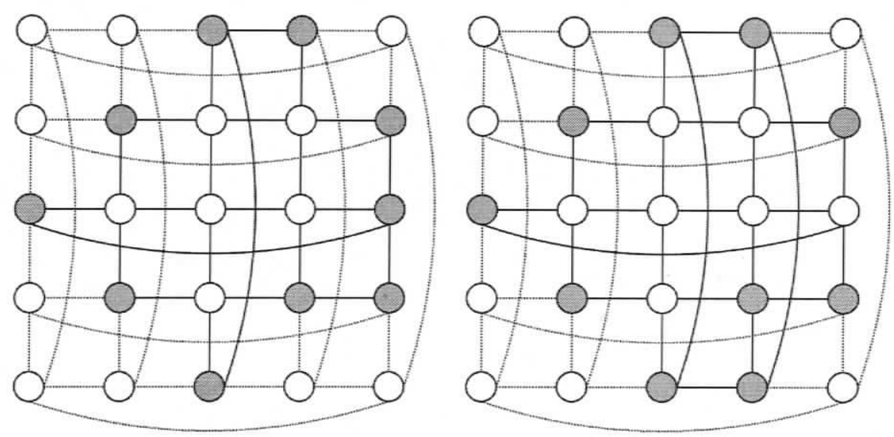
  The case where h is even is a little more complex. We proceed as follows.

- Using the same technique as described in the odd case, we build a diamond of dimension $h/2$. As just described, that process uses $4(d-1)+2$ guards, i.e. $2h-2$ guards and leaves $2h-4$ guards on the exposed nodes of the diamond.

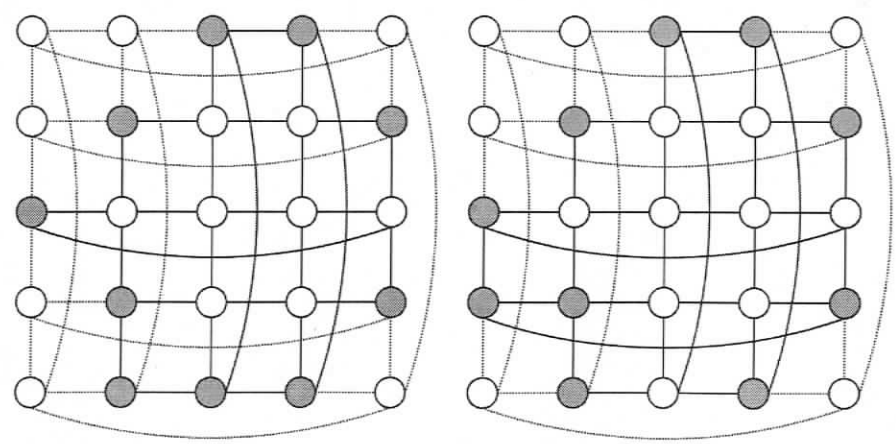
- We then expand the diamond of dimension $h/2$ to form an "asymmetrical diamond" which has three sides of dimension $h/2$ and one of dimension $h/2+1$. The process is illustrated in Figure 3.39, and our description refers to that figure. We note that we have four "extra" guards to play with. For one of the



(a) (b)



(c) (d)



(e) (f)

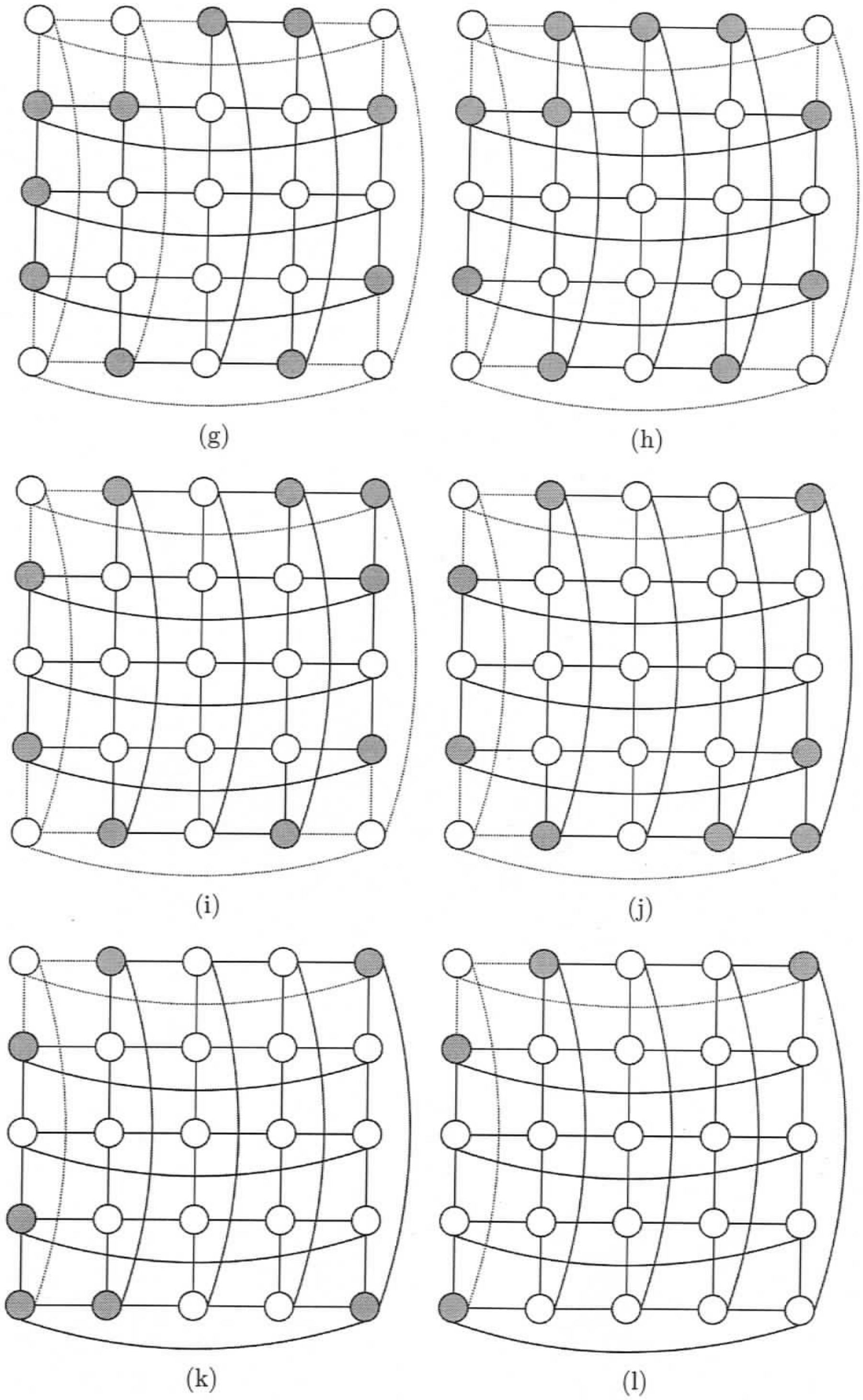


Figure 3.38: Cleaning the corners in the odd case

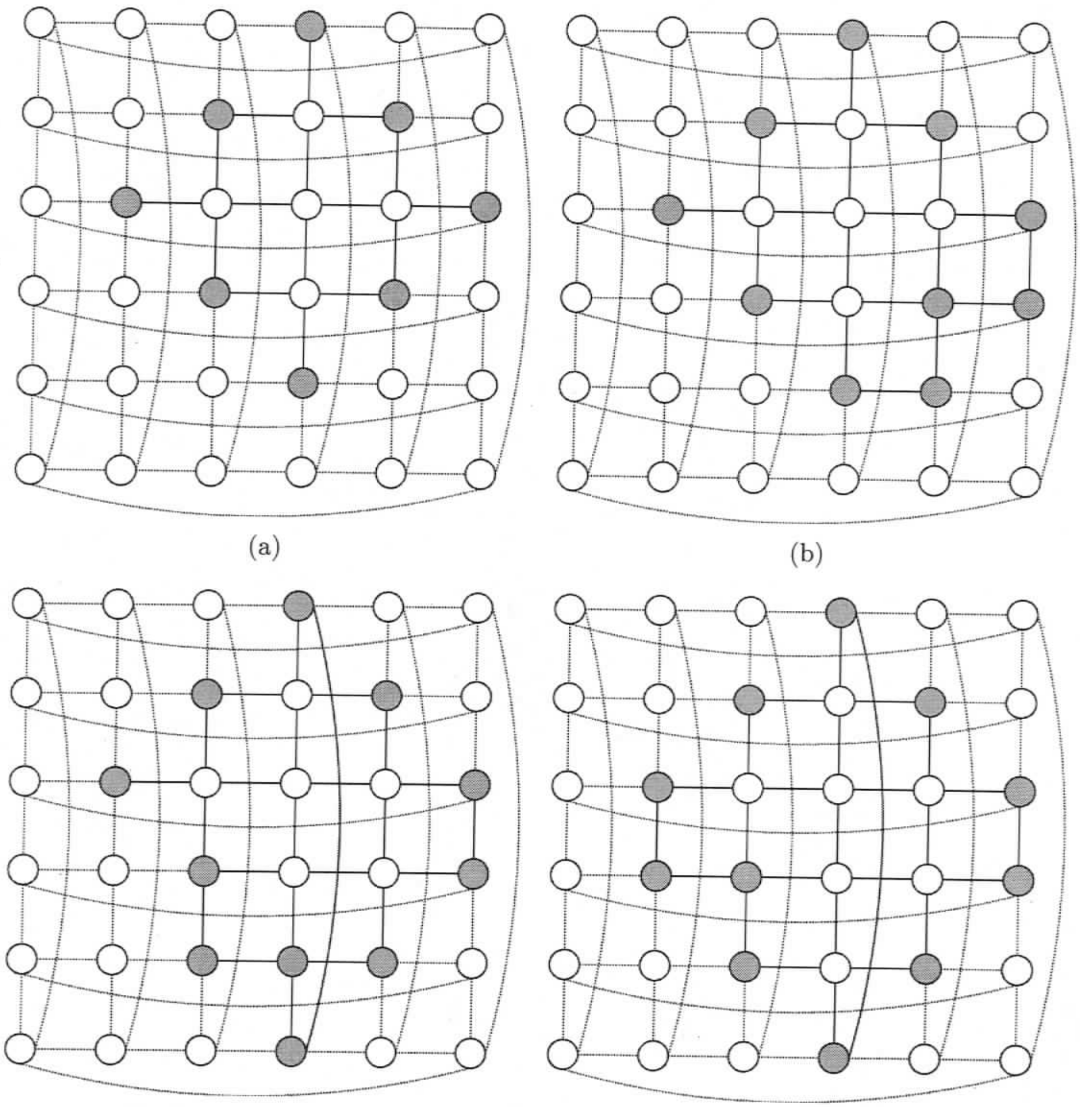
sides, both apex nodes are in the perimeter of the diagram. Our goal in this step is create the asymmetrical diamond with all apex nodes in the perimeter of the diagram.

We start with two new guards which release the guard on the node below the rightmost apex node. The released guard is used to release another guard and so on until the bottom apex is reached. At that point another of the extra guards is necessary. We then proceed along the lower left side without needing any extra guards, until we reach the leftmost apex. There we use our last extra guard and proceed along the upper left side to the top apex. The last move is the removal of one released guard which we will use in the next phase.

At the end of the process there are $3h/2 + h/2 - 1 = 2h - 1$ guards on the graph, leaving one to continue the next phase.

- In the last phase we reduce the diamond to one with "short" non-overlapping sides, such that the number of spare guards is obviously sufficient to complete the cleaning. The entire process is illustrated in Figure 3.40.

  At the beginning of this phase, we have only one extra guard to use, but we observe that there are two exposed nodes that are incident with only one dirty edge. In the figure, we start by releasing the top and leftmost guard, and proceed down the upper right side releasing and then using guards. Note that when we place a guard on the node immediately above the upper guarded apex node, two guards are released. We use both of them to start the cleaning of the lower right side. The bottom apex is incident to just one dirty edge, when it is reached, so one guard is sufficient to release its guard. When a guard is placed on the node immediately below the rightmost apex node, two guards are released. Both of these are necessary to permit the cleaning of the upper left



(a) (b) (c) (d)

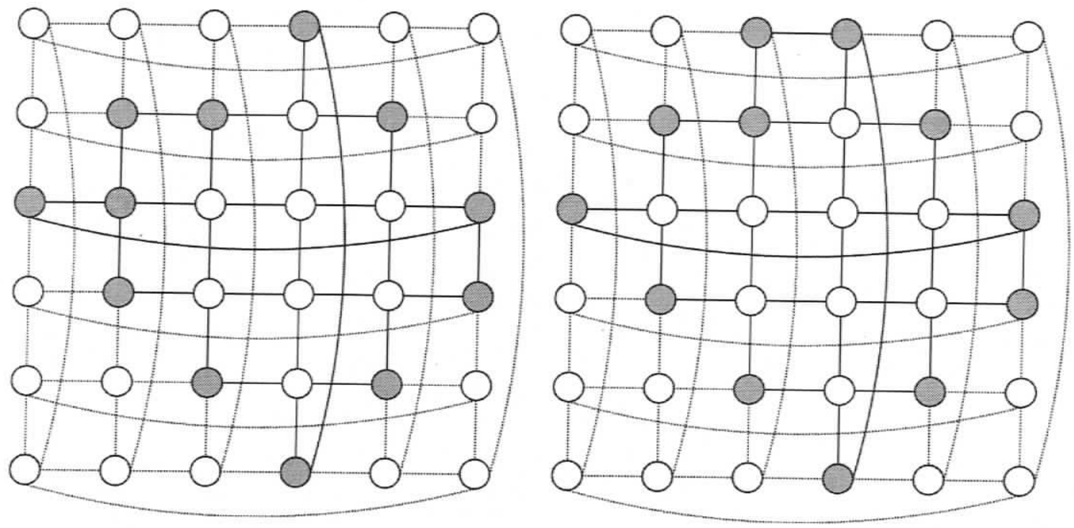


Figure 3.39: Creating an asymmetrical diamond

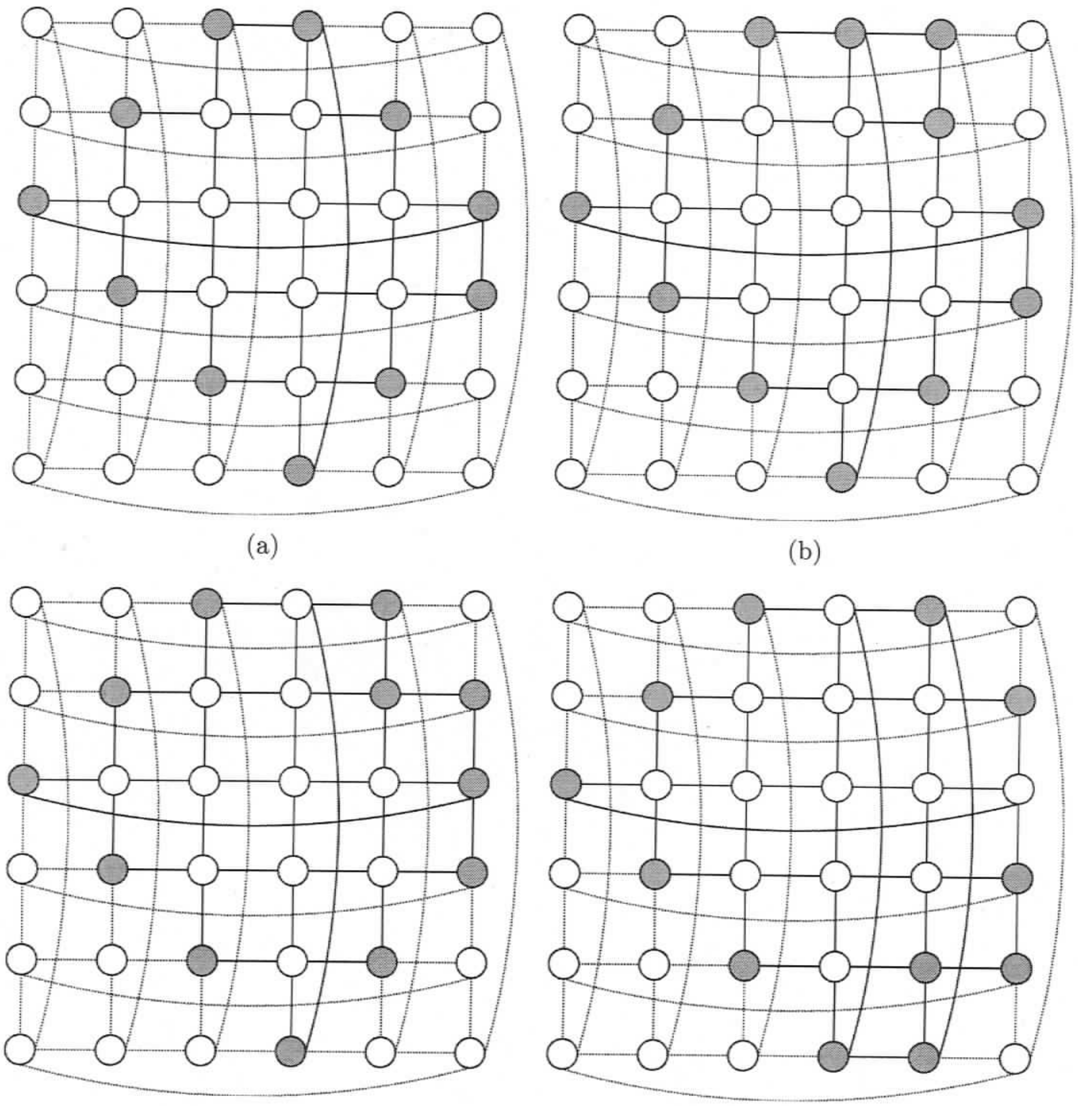
side.

So we have used $2h$ guards to reach a state with four non-overlapping sides, three of size $h/2-1$ and one of size $h/2$. The remaining cleaning can be done in a similar manner, starting now with three extra guards. □

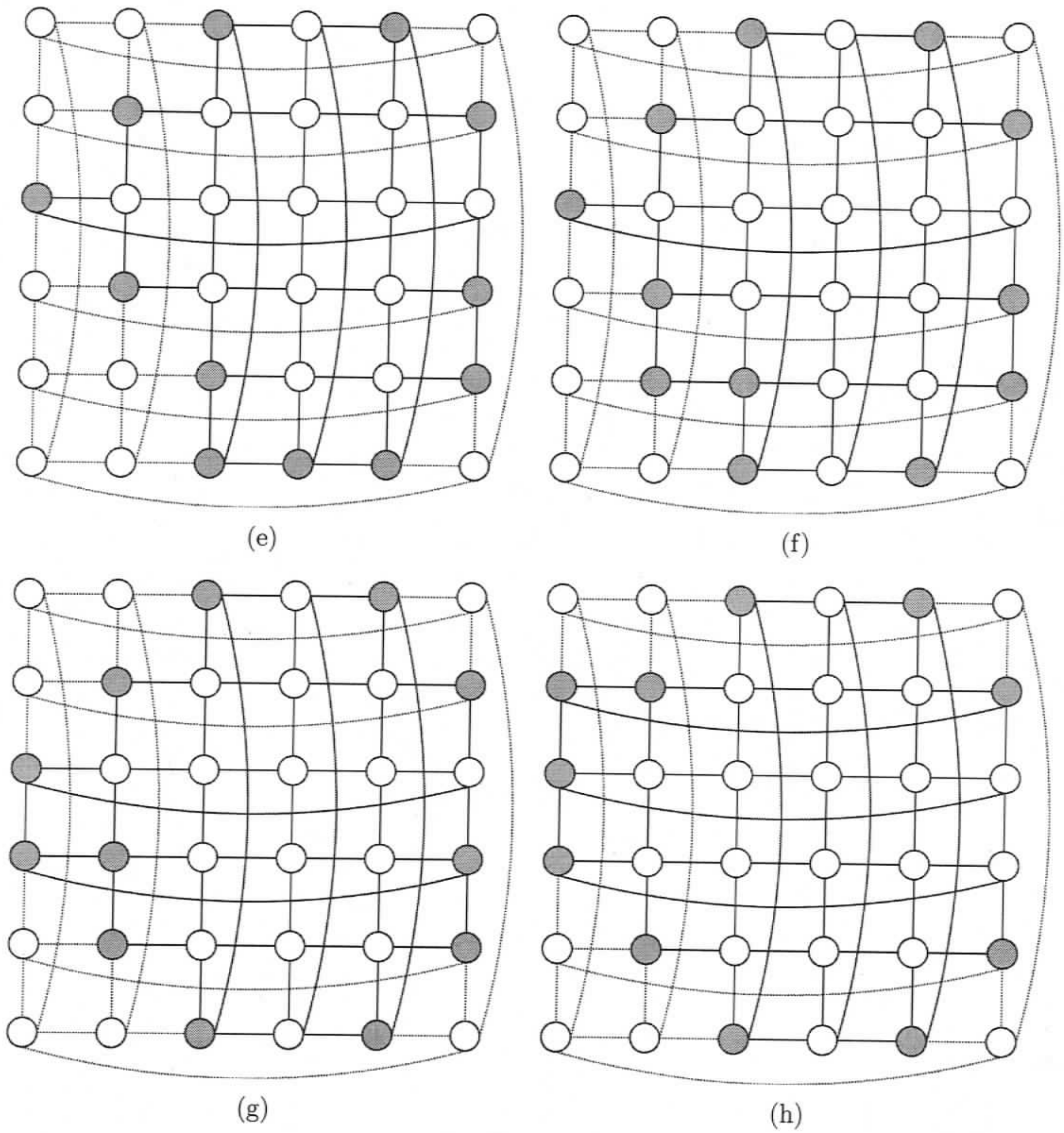
We continue by showing that $2h$ is also a lower bound on equi-dimensional tori.

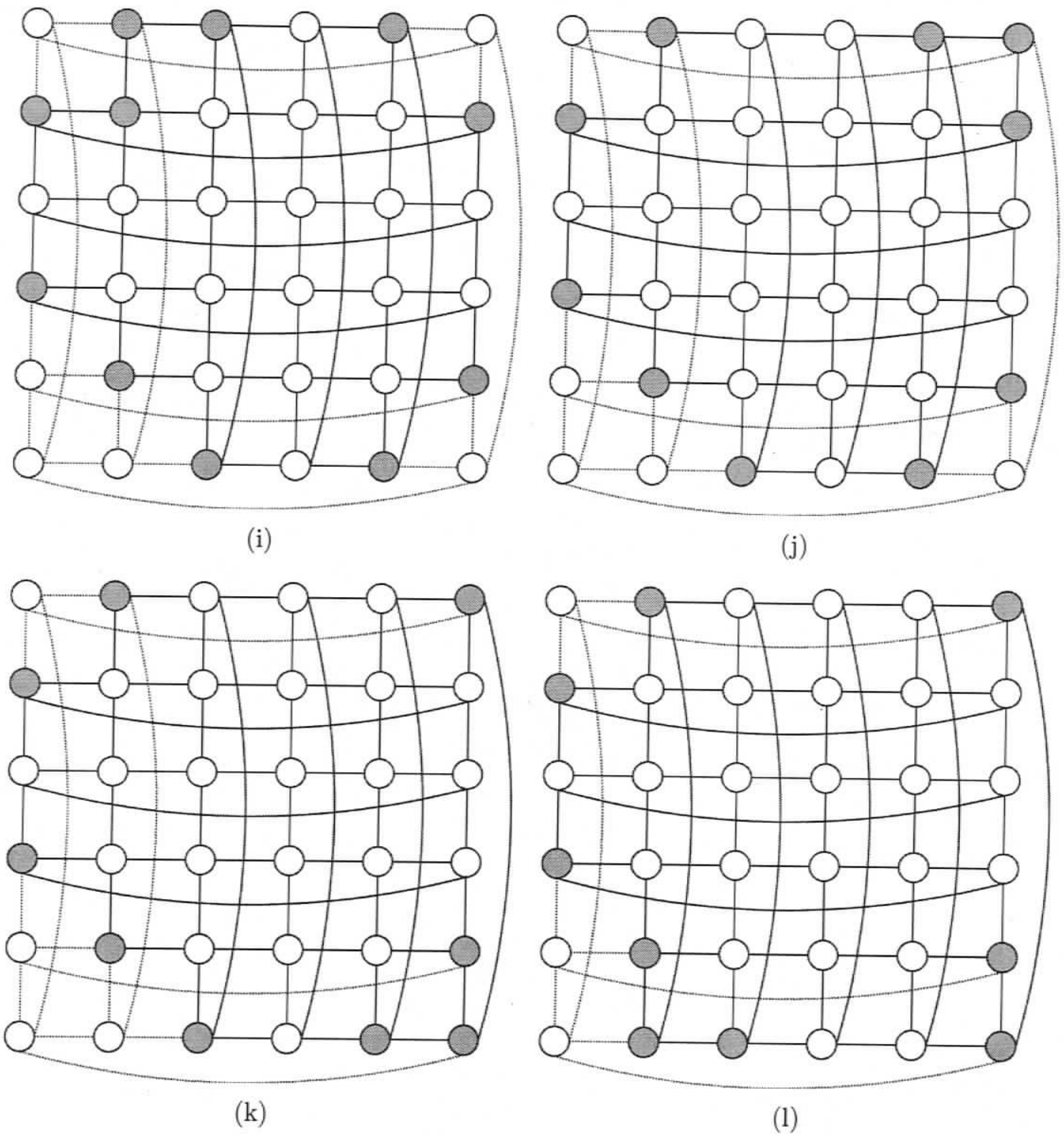
Lemma 33. *If G is an (h,h)-torus, then $ns(G) \geq 2h$.*

Proof. Let step t be the first step at which the set of clean edges defines a component C^t which includes two columns or (inclusive) two rows. Suppose C^t contains two columns, say H and H'. At the completion of step $t-1$ there is one completely clean column, say H, one almost clean column, H', and no more than one completely clean row. Let R be the row where a guard is placed at step t to clean H'. It must be the case that, at step $t-1$, H' contains exactly two dirty edges that are incident with a node v that receives a guard at step t. Further, R cannot be completely clean since v



(a) (b) (c) (d)


(e)
(f)
(g)
(h)



(i) (j) (k) (l)

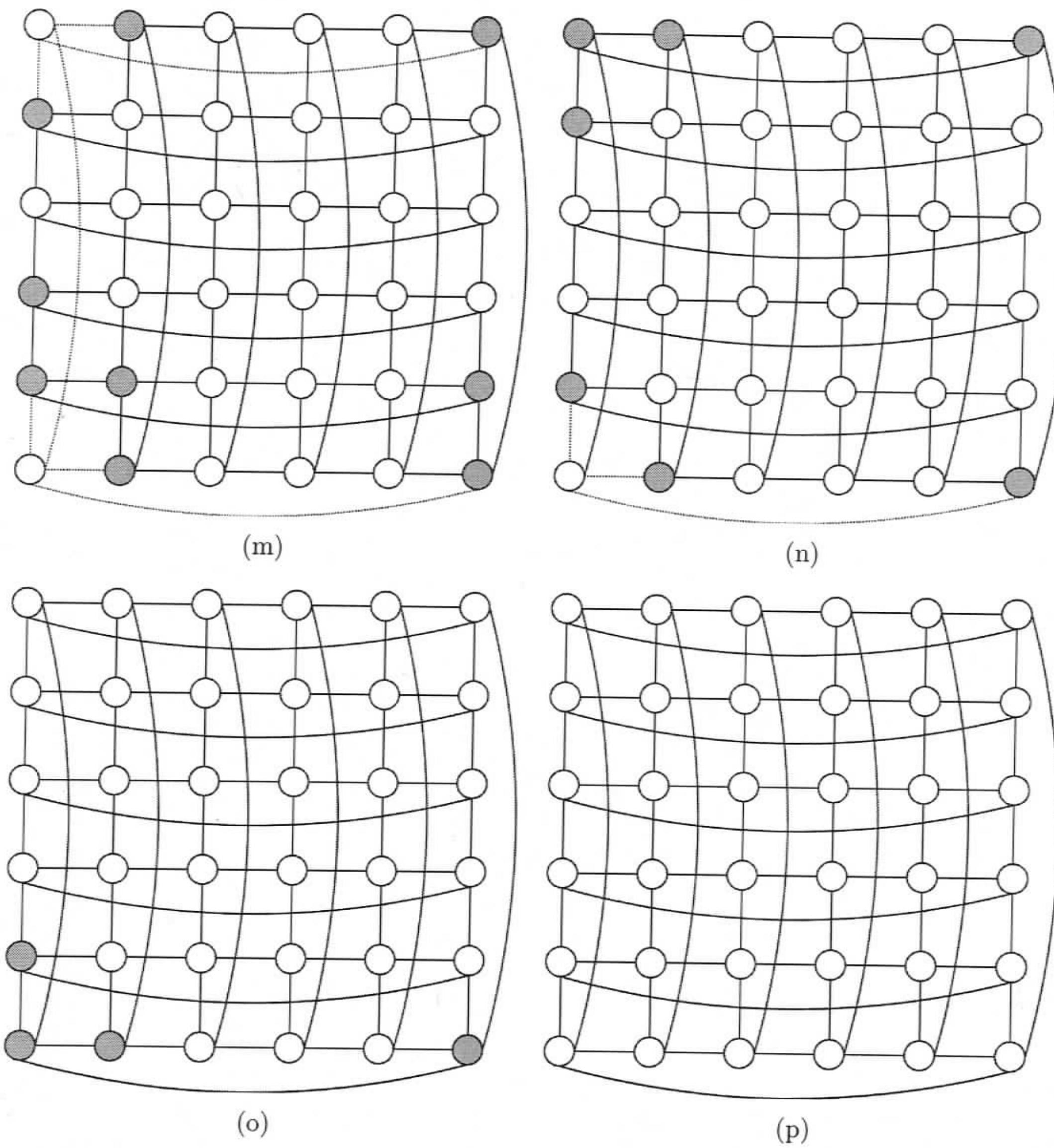


Figure 3.40: Cleaning the corners in the even case

would then be incident with both clean and dirty edges and unguarded. We consider two possibilities, either there is a completely clean row at step $t-1$ or not.

- Suppose there is a clean row at step $t-1$, say R'. Note that R' cannot be the same row as R because R is not clean until t.

  Consider all the rows other than R and R'. Any partly dirty row must contain at least two guards. Any completely dirty row contains two nodes incident with both clean edges (in the clean columns) and dirty edges (in the dirty rows). So each such row must contain at least two guards.

  Row R is not completely clean, but it intersects column H. Hence it must contain a guard. R' is the only completely clean row, hence it is adjacent to a row containing at least two adjacent dirty edges, incident with an unguarded vertex. Hence, R' must contain at least one guard adjacent to that unguarded vertex. Hence, there are at least $2h-2$ guards on the graph at the completion of step $t-1$.

- Suppose that at step $t-1$ there is no completely clean row. All the rows other than R are either partly or completely dirty and, by the reasoning already given above, must contain at least two critical guards. Hence, there are at least $2h-2$ guards on the graph at the completion of step $t-1$.

So, no matter whether a completely clean row exists at step $t-1$, there are at least $2h-2$ guards on the graph at step $t-1$. If there are more than $2h-2$, we are done, so let us suppose there are exactly $2h-1$.

We have shown that R is not completely clean. If it is partly dirty then it must contain at least two guards and hence there are at least $2h$ guards on the graph at step $t-1$ and we are done. If R is completely dirty, consider the guard placement to node v in R at step t. This guard cleans no row edge in R and so is critical.

By the criticality principle, since step t is a placement, all $2h-1$ guards on the graph at step $t-1$ are critical. Both guards in H' adjacent to v are adjacent to at least one dirty row edge at step $t-1$, and the placement to v cannot change this. Hence there are at least $2h-1$ critical guards on the graph at step t. Hence, by the criticality principle, step $t+1$ must be a guard placement, leaving at least $2h$ guards on the graph.

Since the graph is symmetrical with respect to rows and columns, if C^t includes two rows, the same result can be obtained using the same argument with rows and columns exchanged. □

Theorem 7. *If graph G is a torus and $h \neq w$ then* $\mathrm{ns}(G) = \min(2h+1, 2w+1)$. *If $h = w$ then* $\mathrm{ns}(G) = 2h$.

Proof. When $h \neq w$, from Lemma 30 we get an upper bound $\mathrm{ns}(G) \leq \min(2h+1, 2w+1)$. From Lemma 31 we get a lower bound $\mathrm{ns}(G) \geq \min(2h+1, 2w+1)$. Thus, $\mathrm{ns}(G) = \min(2h+1, 2w+1)$.

When $h = w$, from Lemma 32 we get an upper bound $\mathrm{ns}(G) \leq 2h$. From Lemma 33 we get a lower bound $\mathrm{ns}(G) \geq 2h$. Thus, $\mathrm{ns}(G) = 2h$. □

Chapter 4

Conclusions and Future Work

From our initial observation that (h, w)-grids have a node search number of $\min(h+1, w+1)$ we set out to prove that this was in fact true by establishing a lower bound, which, as was expected, turned out to be identical.

Then we expanded upon (h, w)-grids to include a wide variety of grid-like graphs. Initially we added simpler variants such as cylinders and orb webs which as we proved, by finding their lower bounds, have node search numbers of $\min(h+1, 2w+1)$ and $\min(h+1, 2w+2)$ respectively. A more complicated addition were the walls which, due to their unusual structure, were very difficult to prove. Eventually, however, we did establish lower bounds of $\min(2h+1, w/2+2)$.

Finally, there were the tori, the only non-planar graph we studied. The tori are unusual in that, as it turned out, they have both an unobvious upper bound as well as a non-trivial lower bound. The node search number of tori is $\min(2h+1, 2w+1)$ unless $h = w$ in which case it is $2h$.

The techniques used to prove the lower bounds for these grids could be applied to other types of grid-like graphs, such as 3D grids. It is likely that the lower bounds of any graph that can be described in terms of "rows" and "columns" can be proven using these techniques.

Bibliography

[BF02] H.L. Bodlaender and F.V. Fomin. Approximation of pathwidth of outerplanar graphs. *Journal of Algorithms*, 43(2):190–200, 2002.

[BGHK95] H.L. Bodlaender, J.R. Gilbert, H. Hafstemsspm, and T. Kloks. Approximating treewidth, pathwidth, and minimum elimination tree height. *Journal of Algorithms*, 18:238–255, 1995.

[BK96] H.L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.

[BKK95] H.L. Bodlaender, T. Kloks, and D. Kratsch. Treewidth and pathwidth of permutation graphs. *SIAM Journal of Discrete Mathematics*, 8(4):606–616, 1995.

[BL91] B. Bollobás and I. Leader. Edge-isoperimetric inequalities in the grid. *Combinatorica*, 4:299–314, 1991.

[BM93] H.L. Bodlaender and R.H. Möhring. The pathwidth and treewidth of cographs. *SIAM Journal of Discrete Mathematics*, 6(2):181–188, 1993.

[DPPS01] J. Díaz, M.D. Penrose, J. Petit, and M. Serna. Approximating layout problems on random geometric graphs. *Journal of Algorithms*, 39(1):78–116, 2001.

[EM04] J.A. Ellis and M.M. Markov. Computing the vertex separation of unicyclic graphs. *Information and Computation*, 192:123–161, 2004.

[EST94] J.A. Ellis, I.H. Sudborough, and J.S. Turner. The vertex separation and search number of a graph. *Information and Computation*, 113:50–79, 1994.

[GGKS95] P.W. Goldberg, M.C. Golumbic, H. Kaplan, and R. Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995.

[Gus93] J. Gustedt. On the pathwidth of chordal graphs. *Discrete Applied Mathematics*, 45(3):233–248, 1993.

[Kin92] N.G. Kinnersley. The vertex separation number of a graph equals its path-width. *Information Processing Letters*, 42:345–350, 1992.

[KP86] L.M. Kirousis and C.H. Papadimitriou. Searching and pebbling. *Theoretical Computer Science*, 47(2):205–216, 1986.

[KT92] A. Kornai and Z. Tuza. Narrowness, pathwidth, and their applications in natural language processing. *Discrete Applied Mathematics*, 36:87 – 92, 1992.

[LaP93] Andrea S. LaPaugh. Recontamination does not help to search a graph. *Journal of the ACM*, 40(2):224–245, April 1993.

[Len81] T. Lengauer. Black-white pebbles and graph separation. *Acta Informatica*, 16:465–475, 1981.

[LL80] A. Lopez and H. Law. A dense gate matrix layout for mos vlsi. *IEEE Transactions on Electronic Devices*, (27):1671 – 1675, 1980.

[LMC00] C. Lucet, J.-F. Manouvrier, and J. Carlier. Evaluating network reliability and 2-edge-connected reliability in linear time for bounded pathwidth graphs. *Algorithmica*, 27:316–336, 2000.

[MS88] B. Monien and I.H. Sudborough. Min cut is NP-complete for edge weighted trees. *Theoretical Computer Science*, 58(1-3):209–229, 1988.