Grasping Graphs

by

Sarah Carruthers
B.Sc., University of Victoria, 2004

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

Master of Science

in Interdisciplinary Studies (Computer Science, Curriculum & Instruction)

Grasping Graphs

by

Sarah Carruthers
B.Sc., University of Victoria, 2004

Supervisory Committee

Dr. Ulrike Stege, Co-Supervisor
(Department of Computer Science)

Dr. Timothy Pelton, Co-Supervisor
(Department of Curriculum and Instruction)

Dr. Yvonne Coady, Departmental Member
(Department of Computer Science)

Supervisory Committee

Dr. Ulrike Stege, Co-Supervisor
(Department of Computer Science)

Dr. Timothy Pelton, Co-Supervisor
(Department of Curriculum and Instruction)

Dr. Yvonne Coady, Departmental Member
(Department of Computer Science)

ABSTRACT

To date, research of computer science education in the elementary classroom has focused on technology-dependent tools like Alice, Scratch, LOGO and LEGO Mindstorms. While these tools seem to have the potential to support learning in accordance with constructionist theory, they have not lived up to expectations. Results of this research, in particular the impact of programming instruction on student achievement, have been weak or mixed. Possible reasons for this are many, including the corresponding *threshold* and *friction* associated with technology-dependent learning. Inspired by a trend of non-technology-dependent instruction of computer science topics, as demonstrated by the success of *Computer Science Unplugged* by Tim Bell, Mike Fellows and Ian Witten, we have chosen instead to investigate the impact of unplugged computer science instruction on Grade Six students. The shift away from programming instruction may also serve to help dispel the myth that computer science *is* programming. Computer science is a broad and diverse field which impacts the lives of all people in a multitude of ways.

It is not yet clear what the best approach is for integrating computer science education into the elementary classroom. One suggestion is to teach computer science topics such that they support other areas of elementary education. For example, students are encouraged to adopt many different problem solving strategies, as supported by

the British Columbia Ministry of Education's K-7 Mathematics Integrated Resource Package (IRP). These strategies include "draw a picture". Graph theory has the potential to support problem solving as a means of representing complex connections and relationships in a clear and concise manner. Alternatively, a standalone computer science curriculum may be appropriate, in the spirit of the Computer Science Teacher's Association (CSTA) "A Model Curriculum for K-12 Computer Science". Whatever the approach, an important, and fundamental, step in making curricular change is to support the need for change with sound educational research. Only then can we hope to earn the support of the stakeholders, such as school districts and teacher education programs, who can make this change a reality.

In this pilot study, we investigate the impact of graph theory lessons in two Grade Six math classes. Because of the small class sizes and somewhat reduced participation rates, the results of this study need to be verified with further, larger scale studies. However, early indications are that Grade Six students are capable of learning graph theory, and applying it when working on mathematical word problems. In some cases, there appears to be an association between students' ability to apply graph theory as one of many problem solving strategies, and the correctness of their solutions to problems.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

False Robotics (Kyra, Heidi, Farrell, Max, Luke, Cedar, Connor, Estegenet, Ollie, Georgia, Isaac, Gryphon, Freeman, and Hayden) for keeping my spirits high.

Ulrike Stege and Timothy Pelton for mentoring, support, encouragement, and patience.

Todd Milford for his love of numbers.

NSERC for funding me with a Fellowship.

Computer science is no more about computers than astronomy is about telescopes.
Edsgar W. Dijkstra

DEDICATION

To my sister, Rebecca Carruthers Den Hoed
and,
all the people in my family who inspired me to be such a geek:
my Dad, Mums, John and my Grandpa.

Chapter 1

Introduction

1.1 Motivation

Living in a small remote Gulf Island community off Vancouver Island I have had the opportunity to teach computer science (CS) as a volunteer science teacher in the local public school, teaching K-8 students. This connection with a community school facilitated my recruitment of the teachers who, by volunteering their time and allowing me access to their classrooms, made this study possible. As a member of the Solving Problems with Algorithms, Robotics and Computer Science (SPARCS) group of the University of Victoria [49], I have taught computer science to students of various ages at summer camps and other outreach efforts. I have offered lessons based on the activities in *Computer Science Unplugged*, by Bell, Witten and Fellows [2], as well as original lessons and activities developed by myself and other SPARCS members, covering: information theory, graph theory and algorithms, binary numbers, error detection and parity, computer graphics, data compression and encryption. My experience has been one of engagement and enthusiasm on the part of the students. What drives me to continue with these outreach efforts are the highly relevant and intelligent questions that participants ask as they try to connect these activities to their lives. Ultimately, it is this experience that lead me to this study. It is my belief that young students, those in elementary grades, are capable of learning many aspects of computer science, and make meaningful connections between what they learn and the world around them.

My background is in the field of computer science, but I have chosen to present this thesis in a format more typical of educational research theses. It is my intention that this document be accessible to education researchers, in the hope that further educational research in this direction can follow.

1.2 What is Computer Science?

If you want to start an animated debate, ask a room of computer scientists for a definition of computer science. Depending on their focus, background and personal position, each individual will have a slightly different view of what computer science is. This debate is not likely to end any time soon.

It has been my experience that it is a common misconception among non-computer scientists that computer science *is* programming. However, computer science is less about computers than commonly believed. Computer science is a broad field, and computer scientists research many different topics. A scan of the research areas of faculty members in a department of computer science may surprise many people outside this area of study. In one major Canadian University we might find: *computer music, machine learning, trends in data networking, traffic management, quality of service, traffic engineering, network design, optical networks, performance evaluation, queueing theory, computer graphics, colour science, image processing, human perception, non-photorealistic rendering, computational aesthetics, computational photography, logic in computer science, cryptography, foundations of security, verification, computational complexity, database and knowledge-base systems, graph theory, audio signal processing, human computer interaction, theoretical computer science, and algorithms.* The list is long and constantly changing as our relationship with technology evolves. Computer science isn't just about computers, but about computing in general.

Computer science is a discipline with different names: computer science, computing science or informatics. In this thesis I will use the term "computer science", not because it is more correct, but simply because it is what I am used to. It is just as correct to refer to it as computing science or informatics.

This leads us naturally to the question: what is computer science? The Computer Science Teachers Association (CSTA) defines CS as "the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society" [11]. They further state that, based on this definition, CS curricula should have the following kinds of elements: "programming, hardware design, networks, graphics, databases and information retrieval, computer security, software design, programming languages, logic, programming paradigms, translation between levels of abstraction, artificial intelligence, the limits of computation (what computers *can't* do), applications in information technology and information systems, and social issues (Internet security, privacy, intellectual property, etc.)" [11].

According to the Joint Task Force for Computing Curricula 2005, a project of the Association of Computing Machinery (ACM), the Association for Information Systems (AIS) and The Computer Society (IEEE-CS), a computer scientist's role is to: design and implement software; devise new ways to use computers; and develop effective ways to solving computing problems. They identify the following areas of study in a computer science program: algorithms, application programs, computer programming, hardware and devices, human-computer interface, information systems, information management, IT resource planning, intelligent systems, networking and communications, and systems development through integration [13].

Of particular interest to me, and of importance to this study, is theoretical computer science, which focuses on theories of computability. One important area of theoretical computer science is graph theory. Graphs (not to be confused with more commonly known graphs like bar graphs or line graphs) are a mathematical construct which can be used to model relationships and connections. A graph is made up of a set of *vertices* and a set of *edges*. Vertices (typically indicated by a dot or circle) can be connected by edges (represented by lines), to indicate a relationship or connection between objects, concepts, or entities (see Appendix A.1) . An evolutionary tree is an example of a graph, where the species are represented by vertices, and evolutionary connections are represented by edges. Another example is a road map, where cities and towns are connected by roads.

Graph theory has its origins in the historical Seven Bridges of Königsberg Problem proposed by Leonard Euler in 1735. Parts of the city of Königsberg (including two islands) are divided by a river, and connected by seven bridges (see Figure 1.1).

Seven Bridges of Königsberg Problem: Is there a way to walk through the city and cross every bridge exactly once?

This problem can be modeled as a graph, with the distinct land masses (the two islands and the mainland on either side of the river) represented by vertices, and the bridges as edges (see Figure 1.2). The curious reader will try this out for themselves. A solution is provided in Appendix A.1.1.

Graph theory investigates properties of graphs such as: paths, cycles, topology and connectivity. Graphs are used to represent data structures in computer programs. For

Figure 1.1: Bridges of Königsberg

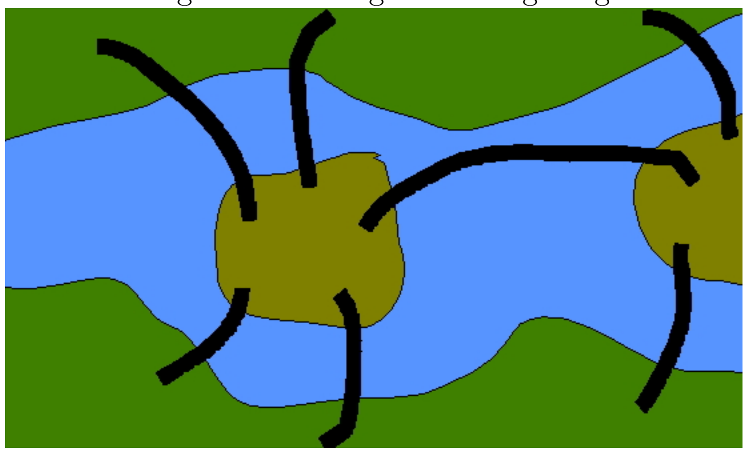
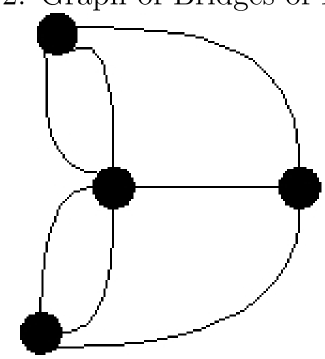


Figure 1.2: Graph of Bridges of Königsberg



example, file systems (which let you navigate and search for files on your computer) are often modeled as a graph, where files and folders are represented by vertices, and edges indicate the hierarchy, or an "is in" relationship. A file "is in" a folder, which in turn "is in" another folder. Findings in theoretical computer science lead to better and more efficient methods for accessing and manipulating files and folders in these file systems. This connection between data structures and graph theory is important. It is an example of one of the many ways in which a sound understanding of theoretical computer science can lead to better programmers and more user friendly computer programs. Graph theory, like many other areas of theoretical computer science, also has connections to other disciplines including biology (modeling habitat and migration patterns), and sociology (analysis of social networks). Finally, graph theory is fundamental to many important areas of study in computer science. If we refer to the areas listed earlier, graph theory is vital to all of them, either in the underlying storage and access systems needed for the data associated with them, or in the representation of models used to define and describe problems.

Success in CS requires many and diverse problem solving and analytical skills which may also support learning and success in other subjects. One of the core components of any computer science education is problem solving. Consider the task of writing a computer program to solve a problem. Let us use a vending machine as a running example.

A first, and fundamental, step in this process is the careful analysis and definition of the problem. In this case, we have a machine, containing some number $0 \leq n$ drinks, which customers can attempt to purchase. If they have enough funds, and there are sufficient drinks, they should be able to make their purchase.

What is the state of the problem at the outset? How many drinks are available? Of which kind? Can the machine make change?

What are the inputs? What denomination coins will be accepted? How do we read them? How does the user select a drink? What if the user chooses to cancel their transaction?

What is the expected output? The machine could output drink(s), change or a refund. How does the machine indicate if it can make change, or if it is out of a drink?

What steps are needed to go from the starting state, given the inputs, to always yield an expected output? The machine must be able to: validate the currency, keep tally of the amount deposited, interpret the user's input (beverage selection, transaction cancellation) and respond appropriately.

How does the machine handle errors? What if invalid currency is inserted? What if the power is interrupted?

Answering these questions is a first step in developing a sound definition of the given problem. It requires a number of skills: analysis, critical thinking, communication, and problem solving. These skills are important outside the world of computer science. In addition to teaching these skills, the inclusion of CS related topics, and computational thinking instruction in the elementary classroom potentially has other benefits. Professionals in various fields will find themselves working with: massive data-sets, networked computers, complex simulations and other CS-dependent technologies. A conceptual understanding of how and why these systems work can allow people to make intelligent choices about what technologies to use.

When we talk about computability in computer science we mean, given a problem and some input, does an algorithm exist which will always yield a correct output.

This definition ignores issues like the amount of time or space (in memory) needed to arrive at the output.

I motivate the definition of computer science with some core questions. These questions are not exhaustive. They are a starting point.

Computer science is the study of computing

- What is not computable?
- What is the most efficient way to compute something?
- What problems are equivalent?
- How do we represent a problem?

A core component of computer science is the study of problems. To a computer scientist, a problem is perhaps somewhat different than it is to the rest of the world, and might be better described as a puzzle or task. There are a number of different classes of problem, including *decision problems* and *optimization problems*. A decision problem is one which can be answered 'yes' or 'no', for example: is there a flight to Mexico City from Vancouver on March 15th? An optimization problem is one in which the goal is to find the best solution, for example: find the cheapest flight from Vancouver to Toronto. Problems have different levels of difficulty or hardness. Consider the difference between the problem of winning a game of tic-tac-toe, versus a game of chess. It is much easier to learn to win at tic-tac-toe, than it is at chess, but why? One way to think about this, is to consider how many possible games, from start to end, there are for each one. If you consider only legal moves, you could try to list all possible moves at each possible turn, and on all subsequent turns. You would realize that there are many more games for chess than there are games of tic-tac-toe. This is one way to gauge the hardness of a problem. Another way is to show that a problem is at least as hard as some other problem. In this way we can devise classifications of hardness. These are just a few examples of the topics of theoretical computer science.

Computer science is also the study of information

- How do we represent information?
- How do we access information?

- How do we share information?
- How do we store information?

We represent information, on various media and devices, for a myriad of purposes. These three variables affect how we answer these questions. Consider sharing a picture with a friend using a mobile device. How is that picture represented on your device? This is at least a two-sided question: how is it represented in memory, and how does the device represent it for the user? How do you access the picture: do you use a keyboard, a touch-screen, a pointing device, your voice? How do you find the picture in the first place? How will it be represented on your friend's device (in memory and on the screen)? How will the devices find each other? How will the devices communicate to transmit the picture? How will your friend know that you sent it, and then how will they access the picture?

So, what is computer science? The more I think about it, the more I find that I prefer a simple definition of computer science, one that is perhaps more spartan than others would like. To me computer science is the study of computing and information. This is a definition that I feel embodies the foundation of all important areas of computer science. If we refer back to the list of topics presented earlier, we can see that each of these topics is in some way a study of either (or both) computing or information. The advantage of a simple definition is that it can remain relevant as our relationship with technology evolves, and as technology itself changes.

1.3 Computer Science and Society

We live in a world of ubiquitous computing. In North America, regardless of geographical location, socioeconomic status or demographic, most youth interact with computational devices on a daily basis: cars, appliances, media devices, communication devices. A large number of youth, 36% of 11-14 year olds, and 56% of 15-18 year olds, own a cell phone. Similarly, 60% of 11-14 year olds and 41% of 15-18 year olds own a handheld video game [44]. Cell phone use appears to be overtaking personal computer use: in 2003, there were 1.4 billion cell phones, compared to 607 million personal computers [24], and, as of 2009, the number of cellular subscriptions worldwide has risen to over 4.5 billion [23]. According to one poll, internet access via mobile device is on the rise, in 2007, 24% of Americans said they had accessed the internet using a mobile device, compared to 43% in 2009 [19].

Because it is often faster and easier to install cellular networks than fixed-line phone networks, mobile phone use is mushrooming in the developing world, rapidly overtaking the use of traditional phone networks [22]. For example, in Cambodia, the ratio of mobile cellular subscriptions to fixed telephone lines is $103.2 : 1$, with $861,500$ cellular subscriptions in 2004 increasing to $5,593,000$ in 2009 [23]. Another trend in developing nations is *shared phone use*, where individuals or businesses lend cellular phones (for a fee) to others who may not be able to afford their own phone [8]. The rapid adoption of mobile technology in the developing world means that ubiquitous computing isn't limited to developed nations. People around the globe have access to handheld computation.

These rapid changes in the way we use technology have many implications. For instance, we now regularly share personal and private information over vast networks. In order to protect personal privacy, it is helpful to understand how these technologies work, at least at a conceptual level. Basic computer science education can provide world citizens with the tools and knowledge necessary to make informed decisions about how to use technology, and how to share information.

But should we teach computer science to elementary level students? At the Western Canadian Conference of Computing Education (WCCCE'10), I was faced with critics who pointed out that while students today ride in automobiles, they are not taught automotive mechanics in elementary classrooms, and therefore the fact that they are surrounded by computational devices does not imply that they need to be taught CS. While elementary science curriculum in Canada does not contain automotive mechanics it does include units in physics and chemistry, the fundamental sciences which provide a foundation for us to understand how and why automobiles work. Similarly, elementary level students ought to be able to learn some foundational elements of how and why computers work. But perhaps more importantly, CS education has broad benefits. Beyond being simply an intellectual pursuit, CS can lead to many career paths, teaches problem solving, supports and connects to other sciences and disciplines, and can be engaging for many different types of learners [11].

It is important that people in fields outside CS be able to communicate effectively with computer scientists, a task made easier with knowledge of basic computer science terminology and technology. Consider, as an example, the study of Orca whale song on the coast of British Columbia [36]. Biologists, using stationary hydrophones located at stations along the coastline, gather countless hours of audio data containing whale-song of one or more passing whales. From this audio data biologists wish to identify specific Orca and, from time and spacial patterns which may emerge, interpret Orca behaviour in a non-intrusive manner. Because of the complexity of this problem, a brute force approach may not be feasible, and through consultation with computer science experts, biologists have a better chance of extracting meaningful data from the audio gathered, in a timely fashion. A basic awareness of CS topics allows these biologists to successfully communicate their specific needs and concerns to their collaborators, and understand the theoretical limitations of the tools they wish to have developed. This will hopefully lead to a better understanding of the behaviour of these intelligent creatures with whom we share this planet.

If we agree that CS is an important component of education the question remains of how CS relates to existing curricula.

1.4 CS and Elementary Curriculum

An understanding of technology, the role it plays in society, and how to use it responsibly are important components of education today. The study of information and communications technology are highlighted in the British Columbia Ministry of Education's Mathematics Grade Six Integrated Resource Package (IRP) as being important to society [34]. In particular, literacy in this area is identified as "finding, gathering, assessing, and communicating information using electronic means, as well as developing the knowledge and skills to used and solve problems effectively with the technology" [34]. The document supports the need for students to be able to understand the ethical issues associated with the use of technology. This parallels nicely with learning outcomes 1-10 suggested for grades 6-8 in the Computer Science Teacher's Association (CSTA) A Model Curriculum for K-12 Computer Science publication [11] (see Appendix A.2).

But even more, computer science instruction has the potential to support learning in general. In learning computer science topics, students learn a number of important skills: problem solving, algorithmic thinking and logical reasoning [11]. According to the IRP, Mathematics K to 7 instruction should integrate the following seven mathematical processes: communication, connections, mental math, problem solving, reasoning, technology and visualization. The integration of computer science topics in math classes has the potential to support a number of these processes: visual abstractions common in computer science can support the communication process; logical reasoning is a fundamental component of the reasoning process; and problem solving, an integral skill in computer science, is fundamental to many aspects of learning.

Computer science is a broad subject, comprised of many different specialized areas of study. A key step in evaluating the teachability of computer science in elementary classrooms is determining appropriate CS topics to teach. In 1975, Niman identified graph theory as a potential CS topic for elementary instruction, due to its versatility in visually representing ideas, and its application to puzzles [33]. Mathematical word problems are an abstraction, but one which the population in question should be familiar with. According to British Columbia's Ministry of Education Mathematics Grade 6 Integrated Resource Package, student should "draw to represent their thinking" when working on mathematical problems [34]. Relational graphs provide a natural way to visually represent relationships and connections between entities, and are a special case of this problem solving strategy. The CSTA Model Curriculum specifically identifies graphs as a CS learning goal for grades 6-8 [11].

It may be feasible, in much the same way that Environmental Education has recently been integrated into the curriculum in British Columbia, to integrate computer science-based topics into the curriculum in such a way that they support many subjects in the elementary curriculum [3, 6].

1.5 CS and the Elementary Classroom

According to Papert's constructionist theory, learning is an active process. Students actively learn through the design and creation of external artifacts [39]. In Kindergarten, for example, students learn by physically manipulating objects. However, as students progress through the grades, opportunities for these hands-on experiences become less frequent. As concepts become more complicated, physical representation of ideas becomes more difficult. Computer programming and simulation, however, provide a means for students to continue to learn through interaction, rather than via more abstract forms [43]. This type of learning, learning through doing, also has the potential to start dispelling misconceptions about computer science, by providing "new things for children to do so that they can learn mathematics as part of something real" [39]. Software packages like Scratch [46] and Alice [1] allow students to create digital artifacts to support learning. Robotics kits like Lego Mindstorms [26] and PicoCrickets [41] allow students to learn through the process of constructing and controlling a physical artifact. The use of simulation software like StarLogo

[51] allows students to control and manipulate complex systems, in keeping with this constructionist approach. Research in computer science education has focused on programming and robotics instruction. While I agree that these types of instruction are appropriate for learning, programming is only one of many areas of computer science which may be suitable for integration into the elementary classroom. Computer science-based activities which are not technology-dependent have also been successfully developed for participants of this age [2], what remains is to evaluate how these types of activities can support elementary curriculum.

1.6 A Collaborative Approach

It is my belief that young students are capable of learning computer science and of finding connections between computer science and the world around them. Computer science has the potential to support and enhance learning existing subjects in elementary curriculum. Graph theory, in particular, appears to be an appropriate topic to support mathematical problem solving. However, changing elementary level curriculum to include and accept computer science as a valid subject requires more than just anecdotal evidence and personal belief. Curriculum reform or change should be informed by sound educational research. The type of research needed to inform curriculum reform is very different from the research typically done by computer scientists, and should conform to standards and best practices already established in the fields of educational, human and behavioural research.

If existing publications are indicative of the current state of computer science education (CS ED) research, then researchers in this area should select studies upon which to base their work with caution [42]. In their analysis of current practices, Randolph *et al.* note a number of shortcomings and flaws found in CS ED research publications, including: flaws in the report elements present as recommended by the American Psychological Association; problems with sampling of participants, with self-selection being the foremost method; and a lack of validity and reliability of measures. Randolph *et al.* also note that many of the studies analyzed used research designs which suffer from weak internal validity.

The approach taken in designing this study was one of collaboration. Rather than reinvent the wheel, the researcher collaborated with experts in the fields of education research to develop a study to extend our understanding of the role computer science can play in the elementary classroom. We recognize that scientific experimentation is not always realizable in the classroom [52]. It is unethical to control many human variables and the many unknown variables present in this type of environment make it unfeasible to deploy true experimental research studies. However, carefully constructed quasi-experimental designs can provide some improvement in the quality of evidence collected.

I have been fortunate to be a part of Pacific CRYSTAL, one of five NSERC funded Centres for Research in Youth, Science Teaching, and Learning, based at the University of Victoria. The CRYSTAL group is a collaboration of educators and education researchers who have been working together to promote science literacy, with a vision of promoting "scientific, technological, engineering and mathematical literacy for responsible citizenship and ecological sustainability through university and community research partnerships" [10]. An important part of this type of work is the dissemination of resources and lessons learned. In order to further work in this area, the Pacific CRYSTAL group is publishing a book written by its group members [5].

This type of collaboration not only benefits the computer science education community, but the computer science community in general, and ultimately the educational community as well. As collaborators, we inevitably learn from each other. In particular, collaboration between computer scientists and education researchers can better inform the education community what computer science is. Computer science is a relatively misunderstood discipline, one which is plagued by misconceptions. A study of high school calculus students found that 80% had no idea what computer science was, and, only 2% had a "reasonably good grasp of what the field of Computer Science entailed" [7] (other scientific disciplines may suffer from similar misconceptions). This misconception is not limited to high school students. Computing literacy courses were described as computer science classes [15]. The Computer Science Teachers Association lists a number of myths about computer science which plague them, including: "computer science equals programming," and "computer science is not a scientific discipline" [11]. Misconceptions about computer science may undermine the efforts of curricular reform, and may contribute to the challenges faced by advocates of the inclusion of computer science in elementary education. By collaborating, as computer scientists, with educational researchers, teachers, school districts and students, we can begin to dispel some of these misconceptions.

1.7 Links to research

1.7.1 Enrolment

A motivation of this study is the apparent discrepancy between the number of computer science graduates, and the job opportunities for computer science graduates. Since the industry's peak around 2000-2001, computer science departments in Universities and Colleges across Canada have seen declines in enrolment [48]. At the same time, computer science graduates emerge into a job market with a higher projected growth level than the average (2.4% vs. 1.1%), and higher than average expected annual earnings ($53,589 vs. $45,157) [47]. While it appears that this trend of declining enrolment may be reversing in the U.S., the same is not yet certain in Canada [56]. Improving enrolment and retention in Computer Science programs would increase the number of computer science graduates emerging from Universities and Colleges to more closely match the needs of the industry.

There are many ways to improve enrolment and retention in computer science programs, including: informing students about computer science, and better preparing students before they enter undergraduate programs [48]. Various approaches for improving high school computer science education are suggested including: supporting CS teacher education, and improving curriculum by better defining what constitutes the CS knowledge base [15]. These efforts are laudable, but elementary math and science curricula should also be examined to identify where elements of computer science can be introduced to support these subjects before high school. As in all fields, we should constantly reflect upon and reevaluate what we are teaching. By strengthening the foundation of computer science education, we should end up with a sounder structure.

1.7.2 Computational Thinking

In recent years the phrase "Computational Thinking" has become a topic of discussion in the education community. In many ways, this has been advantageous to the computer science education community, as it has brought awareness of computer science to educators in other areas. In particular the phrase highlights computing in our discipline, as opposed to computers. According to Jeannette Wing, computational thinking is as fundamental as the *Three R's* (reading, writing and arithmetic), and should be part of every child's analytical ability. In it she outlines what computational thinking is, and how it can support learning [55]. The following is a summary of her definition of computational thinking.

Computer science is a diverse field encompassing: algorithms, networks, informatics, encryption, systems, artificial intelligence, human computer interaction, and many other areas of study. One fundamental area of CS is computability. According to Jeannette Wing, computer scientists ask the following questions:

- What is computable?
- How much time/space is required to compute this problem?
- What is the best way to solve this problem?
- How hard is this problem?
- What problems can humans solve better than computers?
- What problems can computers solve better than humans?

Wing points out that the answers to these questions impact the work of statisticians, biologists, economists, physicists and more. Economies are influenced and affected by countless variables: weather systems, political policies and alliances, industrial practices and advances. These factors span multiple nations and continents and are mediated by communications technologies. Economists must think about these complex systems, and use tools which rely upon ever-evolving infrastructures and technologies. Or consider instead emergency response systems, which rely upon communications systems to relay information in order to minimize: damage to properties, and loss of life. In order to minimize response time, vast computational power is needed to: predict earthquakes, track hurricanes, deploy aid, communicate over vast distances using satellites. The scientists who develop these systems must be able to think and reason in a systematic way. This type of thinking is present in all of these fields, and computational thinking connects it with the theoretical constructs from computer science that define it [55]. These fields impact the quality of life of all citizens of the world.

1.8 Purpose of Study

The overarching questions which motivate this study are: at what age can we successfully integrate CS topics in the elementary classrooms? In what way can these CS topics support elementary curriculum? How can we best integrate CS into existing curricula?

1.8.1 Research Hypotheses

For this study, two Research Hypotheses were distilled from the questions listed above.

Research Hypothesis I: Grade Six students can learn to adopt relational graphs as one of many useful problem-solving strategies.

Research Hypothesis II: The use of relational graphs to solve mathematical problems positively impacts students' mathematics achievement.

1.9 Overview of Thesis

Now that we have motivated this thesis, we give a brief overview of the remainder of the document.

Relevant computer science education research is reviewed in Chapter 2. We highlight studies which have investigated the impact of computer science education on student: achievement, attitude and problem solving. Student comprehension of theoretical computer science topics is reviewed, as well the impact of unplugged computer science education. Finally we look briefly at the cognitive consequences of computer science instruction.

Chapter 3 describes the methodology used for this research study, including the population, instruments and intended analysis techniques. In this chapter we also give an overview of the five graph theory lessons.

Chapter 4 presents the data collected during the study.

In Chapter 5, we discuss the results of the study. In particular we focus on how the results speak to the research hypotheses. Post hoc analysis of student engagement/opportunity is discussed. Finally we take a first look at student representations of social networks, and propose how these preliminary results can serve future research. In this chapter, we also discuss the limitations of this study, with a focus on how we can best learn from the problems which emerged.

Finally, some conclusions are presented in Chapter 6.

Chapter 2

Literature Review

Elementary-level computer science education research to date has focused on evaluating the impact of programming and robotics instruction on student achievement, problem solving and attitude. Computer science education is important, and while programming is an important part of computer science, it is not the only component of computer science which can be integrated into elementary curriculum. Furthermore, some elementary teachers are reluctant to include technology-dependent computer science activities in their classrooms [52]. Strained IT budgets, antiquated computer labs, and limited teacher familiarity with computers in general, may influence teachers' choice to avoid computers in the classroom. For that reason, the focus of this study is on "unplugged" computer science activities. Theoretical computer science topics lend themselves well to this type of instruction, as shown in *Computer Science Unplugged* by Tim Bell, Mike Fellows and Ian Witten [2]. However, little research has been conducted to evaluate the impact of theoretical computer science instruction on student outcomes. In addition, very little research has been conducted to investigate young learners' mastery of theoretical computer science constructs. Finally, it is also important to understand the cognitive consequences of computer science instruction.

2.1 The Impact of Programming Instruction

According to Papert's theory of constructionism, people learn through the process of creating tangible objects or artifacts [39]. In *Technologies for Lifelong Kindergarten*, Michael Resnick identifies the importance of computer-based instruction for young learners from a constructionist perspective, as it provides a means by which learners can actively create external artifacts [43]. The process of design and creation of these artifacts allows students to actively construct knowledge and, in the process of creation, learn. This is reflected in Papert's slogan: "Children learn by doing, and by thinking about what they do" [37]. Computer programming and robotics can provide a hands-on learning experience in which learners can manipulate artifacts, either virtual or physical.

A number of studies have been done to evaluate the impact of computer science (most frequently programming or robotics) instruction on elementary grade students' problem solving skills and/or performance on standardized achievement tests. The results of these studies have been generally weak or mixed. Programming instruction was found, in some cases, to have little or no impact on problem solving skills [27, 29], or to impact the problem solving skills of only a subset of participants [30, 50, 53]. Some studies found that programming instruction had no significant effect on the problem solving skills of "average classes", but positively impacted the problem solving skills of the most talented students [30, 32].

Studies examining student success on standardized achievement tests following computer-based instruction on a CS related topic showed no significant difference between the scores of the experimental and control groups [29, 50, 53]. Lindh and Holgersson found that LEGO robotics instruction had a positive impact on performance on maths tests, but only among medium score students. No significant effect was observed on high or low score students [29].

LOGO [31], a programming language designed for young learners, has been the focus of a number of studies of the impact of programming instruction on elementary and middle school students. An early study of the impact of programming on young children's cognition found that LOGO instruction had a significant effect on participants' fluency, originality and divergent thinking. Children who received LOGO instruction were more likely to produce original and creative ideas [9].

With the emergence of the microcomputer in the early 1980's, there was an expectation that programming instruction would have a positive impact on student problem solving and achievement. However, nearly 30 years later, it is not clear that this type of instruction has lived up to this expectation. A possible reason for the failure of programming instruction to meet expectations is that technology-dependent teaching often has high levels of *threshold* and *friction*, which can inhibit adoption [40].

However, this apparent failure is not reason to give up, as there appears to be a positive connection between programming instruction and student attitudes towards learning. Computer science instruction, in particular programming, has been linked to improved student attitude towards science learning. Lehrer *et al.* [27] found that students who were taught to use LOGO to solve geometry problems outperformed both those students who received training in LOGO to solve non-geometry problems, and those who had not received LOGO training. Programming instruction was found to have positive results on student planning [27]. The use of robotics was perceived by teachers to make science more interesting to learn, and allowed for more practice with problem solving [45]. Papert also cited LOGO's benefit of "concretizing the students processes of learning and accomplishments" [38].

2.2 Unplugged CS Instruction

Little research has been done to date to investigate the impact of unplugged computer science instruction on elementary grade learners. A recent pilot study indicates that young learners can be taught recursion using unplugged activities [16]. There appears to be a growing interest in unplugged computer science activities, as evidenced by the interest in and activity on the *Computer Science Unplugged* [2] website, where new activities are added regularly.

A recent study of children in Grades Six to Eight found that instruction of unplugged computer science topics (including deadlock, concurrency and recursion, detailed below in Section 2.3) resulted in positive student attitudes towards computer science, and that participants generally enjoyed the experience. Participants were able to identify that communication and cooperation were needed in order to succeed in the given problems, and were able to come up with creative solutions to complex tasks [17].

2.3 Understanding of Theoretical CS Concepts

Another important task is to investigate how elementary students at various stages of their development understand core concepts in theoretical computer science. Niman suggested in 1975 that graph theory, in particular, would be an appropriate topic to teach at an elementary grade level and proposed a series of lessons to introduce basic concepts [33], but it appears that no follow-up research has been done to support or deny this conjecture. A study was conducted in England to determine how 13 and

14 year olds with no prior instruction in logical implication applied logic in solving problems [20]. It was found that students had a very limited understanding of the logical implication, and often failed to have confidence in what they did understand. A recent study found that Grade Six to Eight students showed some understanding of–and ability to apply–recursion when programming [16, 17]. In particular, students were able to grasp visual recursion, indicating that this may be an appropriate motivator in teaching this complex concept. This same study investigated young learners' ability to understand two computer science concepts: deadlock, and race conditions.

Deadlock occurs when entities require access to a shared resource (which they may modify in the process). If entities need the shared resource, they can request it, and *lock* the resource to prevent another entity from accessing it. If two or more entities are waiting for each other to release the lock, deadlock occurs. Each entity will potentially wait forever for the other to release the *lock*. Participants in the study were able to comprehend deadlock [17].

A *race condition* occurs when entities are modifying a shared resource and the order in which the modifications occurs impacts the output. Imagine two people who are simultaneously editing a shared document. If one person is editing a paragraph while the other is deleting it, and both individuals save the file at *exactly* the same time, what is the end result? Initial results indicate that the concept of race conditions is a challenge to teach young learners [17].

2.4 Implications of Past Efforts

While at first glance programming may be a natural choice for computer science instruction, it has failed to live up to early expectations in terms of having an impact on student achievement. Computer science is a diverse topic, and there are other, potentially more appropriate, manners in which it can be taught in elementary classrooms. As such, an unplugged approach is suggested for this study, specifically: graph theory instruction. In order to validate the appropriateness of any curricular change, sound educational research is needed. Informed by lessons learned from prior work in computer science education research, the next chapter describes in detail the research method chosen to investigate the impact of computer science instruction in an elementary classroom.

Chapter 3

Methodology

This study used a Pretest-Posttest design with non-equivalent groups, an approach classified as a quasi-experimental design [14]. Because the study used intact classes, participants were not randomly assigned to treatment or control group. Treatment groups participated in five graph theory lessons, which included: instruction, discussion, group exercise and individual exercises. The lessons made use of iClicker technology to facilitate discussion and to promote "just in time" teaching.

3.1 Participants

This study took place in two middle schools in a town on Vancouver Island. According to the District Principal of the School District, the socioeconomic status of students at the schools differed. No specific details were given regarding these differences, but based on this information one treatment and one control group were assigned at each school.

Four grade six classes participated in the study: two treatment and two control. The classes at one school will be identified as treatment one (T1) and control one (C1), and the classes at the other school will be identified as treatment two (T2) and control two (C2). These groups were further divided into subgroups, hereafter referred to as: T1a & T1b, C1a & C1b, T2a & T2b, and C2a & C2b (details about why these groups were divided are provided in Section 3.4.1). Because the treatments took place as part of the regular math class, students were not randomly assigned to treatment or control groups, rather, at the request of the District Principal, one teacher at each school volunteered their class as treatment group. A total of 79 students participated in the study (45 male, 34 female). C1 consisted of 9 participants (4 male, 5 female), T1 consisted of 20 participants (8 male, 12 female), C2 consisted of 22 participants (15 male, 7 female), and T2 consisted of 28 participants (18 male and 10 female). See Table 3.1. Class T2 and C2 each consisted of 30 students. Class T1 consisted of 27 students, and C1 of 26 students. All students in the class participated in the treatments regardless of whether or not they were participating in the study. There were a total of 305 grade six students (149 male, 156 female) at the school from which groups T1 and C1 were selected. There were a total of 300 grade six students (177 male, 123 female) at the other school.

Table 3.1: Participants by Group

Group	**Male**	**Female**	**Total**
C1	4	5	9
T1	8	12	20
subtotal	12	17	29
C2	15	7	22
T2	18	10	28
subtotal	33	17	50
Total	45	34	79

3.2 Materials

3.2.1 Instrument

The same test was used for both pre- and posttests, for all groups. This test will be referred to hereafter as the Graph Theory Test (GTT). The GTT consisted of four questions, each a math word problem appropriate for Grade Six students based on the Ministry of Education's Prescribed Learning Outcomes [34, 35] and the researcher's experience with similar problems in the classroom. Each question could be solved with or without the use of Graph Theory and was presented on its own on one side of a page. Blank space was provided on each page for students to work on the problem should they choose to draw a picture or do calculations (see Appendix A.4).

This test was not piloted prior to use in this study. However, Questions 2 and 3 were based on activities which had been piloted with students in Grades Four through Eight, inclusive, and were chosen for the test based on the researcher's experience with the success of other Grade Six students during these activities. The validity of the test, as a whole, is not certain. As all tests were coded by the researcher, the reliability of the GTT is not known. In order to ensure consistency in scoring tests were scored according to a rubric (see Appendix A.6). In the future, the reliability of this measure could be addressed by having student solutions scored according to the rubric by other parties, and having those scores compared to the researchers coding.

The first question on the GTT, Q1, asked participants to draw a picture of a series of friendships (given in the question text), and then answer a series of questions about these friendships/connections. In Q2, students were provided with a table which listed the costs to fly between a selection of Canadian cities. Students were then asked to identify the best route between two cities, namely the one with a minimum number of stops, and calculate the total cost for a family to travel on this route. This is equivalent to a "shortest path" problem (see Appendix A.1). The third question, Q3, asked for the cheapest way to connect a series of Canadian towns, essentially a Minimum Spanning Tree problem (see Appendix A.1). Finally, Q4 asked students to find the most cost effective way for a group of students to share calculators, given their math class schedule.

As mentioned earlier, abstraction in mathematical word problems is appropriate for this level of student. There were two main levels of abstraction in the problems presented in the GTT used in this study. In some cases, the word problem describes the entities and relationships which the student could then use to create the vertices and edges, respectfully, in a graph. In another case, the problem describes the entities which can be used to generate the vertices of the graph, but not the edges. These differing levels of abstraction were chosen to investigate if there is a difference in the participants ability to handle different levels of abstraction when using relational graphs to solve mathematical word problems.

Questions 1-3 on the GTT were of the first type, with two minor variances in abstraction within this group. Question Three (Q3) was the least abstract question, as the relationships in the word problem between the entities could be mapped directly onto physical routes connecting cities and towns in British Columbia. Question Two (Q2) was similar in that it dealt with routes between places, however the routes in this case didn't map directly onto physical objects, but were paths that planes would fly between Canadian cities. Question One (Q1) described relationships between people and, like Q2, while the entities can be mapped onto fictional characters, the relationships between these characters (in this case friendships) cannot be mapped directly onto a physical object, but rather onto the idea of what constitutes a relationship.

For the final question, Q4, the application of graph theory to solve the problem involved a higher level of abstraction. In this problem, the schedules of seven students' math class are given, along with a set of resources that they wish to share as efficiently as possible. This type of problem can be solved with graph theory using a *conflict graph*. A *conflict graph* is created by connecting entities which conflict in some way, and then finding the minimum colouring of the graph, where the colours of the vertices in the graph represent the shared resource(s). The level of abstraction is different for this problem, as the connections which map onto edges in the graph are not given directly in the question, but rather must be extracted from the text.

3.2.2 Treatments

All five treatments for both treatment groups were taught by the researcher. The same lesson plans were used for the two treatment groups, and the treatments were done on the same day,. The treatments consisted of five one hour lessons, over a period of five weeks (November/December 2009). These five lessons covered: basic graph theory, properties of graphs, problems and algorithms, and applications of graphs. For exact lesson plans, see Appendix A.7.

3.2.3 iClickers

This study made use of iClickers [21], a student response system which allows students to answer questions anonymously in class using a remote control. The system also allows the instructor to display the responses in the form of a bar graph or pie chart either in real time or with a delay. Student responses during a session can be saved and analyzed at a later date. There are a number of advantages of using this type of technology in the classroom. It appears that anonymous response systems can elicit engagement from students who might not otherwise respond. Feedback to student responses can be very quick, and quick feedback time has been linked to higher test scores under some circumstances [25]. Where feasible, the interventions made use of iClickers to promote active, engaged learning. This can be a useful way to engage students who might otherwise be unwilling to make their voice heard.

3.3 Data Collection

Approval for this study was given by the Human Research Ethics Board of the University of Victoria Approval from School District 69 was negotiated through the District Principal who reviewed the researcher's proposal (see Appendix A.3) and ethics approval, then recruited teachers to participate in the study.

Pretests and posttests were administered by the teachers in the treatment and control groups during regular class time. While it would have been preferable to have had the researcher administer the tests for consistency, this was not feasible due to constraints on both the teachers' and researcher's time. To ensure anonymity, teachers assigned identification numbers to participants for use on all tests.

3.4 Procedures

3.4.1 Research Design

The study was designed as a quasi-experimental research study, with a single pretest, and two posttests completed by all participants in both treatment and control groups. When participants write the same test more than once, there is a chance that the test itself may affect the results of the study. One way of accounting for this impact is to use the Solomon Four-Group Design.

In a typical Solomon Four-Group Design, participants are randomly assigned to four equivalent groups: two treatment groups and two control groups. One control and one treatment group receives a pretest, and a posttest. The remaining control and treatment groups receive only a posttest. Both treatment groups receive an intervention (see Table 3.2).

Table 3.2: Solomon Four-Group Design

Group	**Pretest**	**Treatment**	**Posttest**
T1	*GTT*	*X*	*GTT*
C1	*GTT*		*GTT*
T2		*X*	*GTT*
C2			*GTT*

The design chosen for this study was a modification of this design, for reasons which are summarized below. The teachers of all four classes, both treatment and control, were asked to select a test of their own choice (TestB) which they could administer as a pretest and as a posttest. They were also provided with copies of the GTT. Each class was divided into equal halves. In the treatment groups, one half (T1a and T2a) would write the GTT, participate in the treatments, write the GTT shortly after all five treatments were completed, then write TestB at a later date, after the appropriate material was covered. The remaining participants in the treatment groups (T1b and T2b) would write TestB, participate in the treatments, write the GTT shortly afterwards, then write the GTT again. The procedures for the control groups were to be the same, except no treatments were administered between the pretest and the first posttest(see Table 3.3).

According to the plan, the pretest was to be administered the week prior to the first intervention, and the first posttest was to be administered the week immediately following the fifth intervention. The second posttest was to be administered, at the teacher's convenience, after the subject matter in TestB was covered in class.

Table 3.3: Modified Solomon-four Design

Group	**Pretest**	**Treatment**	**Posttest 1**	**Posttest 2**
T1a	*GTT*	*X*	*GTT*	Test*B*
T1b	Test*B*	*X*	*GTT*	*GTT*
C1a	*GTT*		*GTT*	Test*B*
C1b	Test*B*		*GTT*	*GTT*
T2a	*GTT*	*X*	*GTT*	Test*B*
T2b	Test*B*	*X*	*GTT*	*GTT*
C2a	*GTT*		*GTT*	Test*B*
C2b	Test*B*		*GTT*	*GTT*

The reason for this modification of the typical Solomon-four design is as follows. The Solomon-four design requires four equivalent groups, two of which receive the treatment, and two of which act as controls. However, due to the socioeconomic differences of the two schools, it was not feasible to have four equivalent groups. Instead, the participants at each school were effectively divided into four groups, two groups each in the treatment and control groups. Rather than have half of a grade six math class sit idle while the remaining half wrote the GTT, it was decided that the remaining students could write a different test, one which was of value to the teachers. The other modification made to the Solomon four design was the addition of a second posttest, allowing the researcher to identify if there was a change in the participants performance on the test after a period of time elapsed following the treatment. Half the participants wrote the GTT, and half wrote TestB, to prevent students from writing the GTT three times, and without having them sit idle during the testing period.

These specific modifications to the solomon four-group design have not been validated, however, modifications to the solomon four-group design have been made in other research areas in order to address ethical or management issues [18, 54].

3.4.2 Intended Analysis Procedures

While it is often common to use analysis procedures designed for continuous data (either interval or ratio) in this type of study, we chose instead to use categorical data analysis procedures. Deviations from the intended research design resulted in reduced counts of data values, making this data less suitable for t-test or ANOVA analysis. Categorical data can either be ordinal or nominal. In this case, GTT test scores were further refined into ordinal scales making them suitable for contingency table analysis. Scoring procedures are described in detail below.

Scoring Procedures

Student solutions to the GTT were scored by the researcher according to a rubric (see Appendix A.6). For each question, a "Draw a Graph" (DAG) score was assigned: 0 for no picture, 1 for a non-graph picture, and 2 for a graph picture. A sum DAG (DAGSUM) score was given (simply the sum of all four DAG scores), and a categorical DAG sum score (DAGSUMCAT) was coded from this value so that DAGSUMCAT $= 0$ if $0 \leq$ DAGSUM ≤ 2, and DAGSUMCAT $= 1$, if $3 \leq$ DAGSUM ≤ 8.

In addition, on each question, the correctness of student responses were scored according to the rubric. Because some questions included more than one correctness portion, an ordinal correctness score was devised ranging from 0 to 2. For Q1 scores for parts 1C and 1D were summed ($0 \leq$ SUM1 ≤ 8) with the resulting categorical score CAT1 $= 0$ if SUM1 < 4, CAT1 $= 1$ if $4 \leq$ SUM1 ≤ 7, CAT1 $= 2$ if SUM1 $= 8$. Parts 2A2 and 2B were summed ($0 \leq$ SUM2 ≤ 4) and the categorical score CAT2 set to $= 0$ if SUM2 ≤ 2, CAT2 $= 1$ if SUM2 $= 3$, CAT2 $= 2$ if SUM1 $= 4$. The same scoring was used for Q3. The 4B score for Q4 was directly as the correctness score. A summary score was generated, across all four questions.

Categorical Analysis Procedures

Contingency table analysis is an appropriate method for analyzing the resultant ordinal GTT data [28]. Two main measures of association are appropriate: McNemar's test, and the Tau measure of association.

In the case of the comparison of participants' "Draw a Graph" (DAG) score before and after intervention, the resultant data is paired polychotomous data. McNemar's test is an appropriate measure of association, as the scores represent two indices of the same characteristic [12]. This test allows for testing of disagreement or change in these variables, in this case as a result of graph theory instruction.

In order to test the second research hypothesis, the association between participants' DAG score and the correctness of their solutions will be tested. For this type of ordinal data, the Tau measure of association is appropriate to determine if there is a relationship between the independent variable and the dependent variable [28].

Post Hoc Analysis

Additional post hoc analysis was done to investigate a number of patterns which were observed in the data. These included student engagement levels on the questions on the pre- and posttests, and drawings in response to Q1. Mean and standard deviations of summary scores were used to investigate trends in achievement. These analysis techniques are described in detail in the Results section.

Chapter 4

Results

A number of factors resulted in a different implementation than originally planned regarding the administration of the GTT for this study. Modifications to the design are described here. General observations during the five treatments are presented in the subsequent sections. Finally, results from participants' GTT scores and answers are presented. Initially, student responses were coded using a scoring rubric (see Appendix A.6). Based on patterns which emerged from the data, further coding of participants' engagement was conducted on a question by question basis.

The small cell counts preclude many of the tests that could normally have been done on this type of data. For this reason, many of the tests are exploratory in nature. McNemar's test was conducted on the participants' DAG score (where DAG stands for Draw a Graph), a categorical score value based on the type of picture drawn while working on the test problems. Contingency analysis was used to compare DAG score to correctness of solutions of each question, using the Tau measure of association due to the ordinal nature of the variables. Mean scores were calculated for the GTT. In response to patterns which emerged from the participants' test answers, some engagement results are presented. Finally some qualitative results are presented for participants' representation of social networks.

4.1 Instrument Administration

A number of factors contributed to modifications to the planned administration of the pre- and posttests.

All four class groups (T1, C1, T2, C2) were arbitrarily divided in half by the applicable teacher. For clarity these groups are labelled *a* and *b*. In all cases, GTT refers to the same test, with the same four questions in the same order. The teachers of all four classes were instructed both verbally and via email how to deploy the pre- and posttests. Group T1 was correctly administered the tests, with subgroup T1a writing the GTT as pretest, GTT as the first posttest after the intervention, and the alternate test as the second posttest, and with subgroup T1b writing the alternate test as pretest, GTT as the first posttest, and GTT again as the second posttest. Two teachers, those of C1 and T2, did not properly administer the first posttest, and T2 also incorrectly administered the second posttest (see Table 4.1 for details). In addition, both control groups, C1 and C2, failed to administer the second posttest to their classes. It is not clear why these posttests were not administered as neither teacher has responded to queries on the matter.

4.2 Treatment: Lesson 1

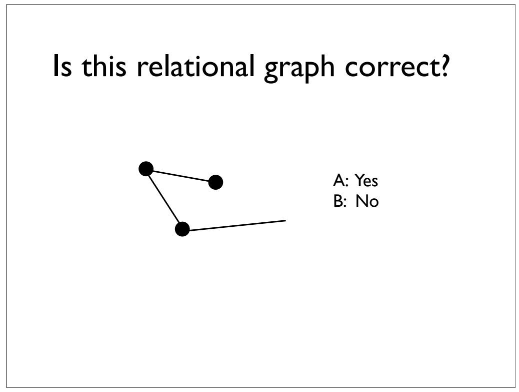
This lesson introduced the students to relational graphs. Students were given five rules for drawing relational graphs correctly (see Appendix A.1.2). Connections were made between relational graphs and: other types of graphs, geometry, and real-world applications. Students were encouraged to discuss things in their lives that relational graphs could convey such as: communication networks and family trees. The second half of the first lesson was comprised of examples of translating ideas into graphs.

The lesson consisted of: instruction, group exercise/discussion, individual work and iClicker [21] activities in which participants were shown a series of graphs and

Table 4.1: Instrument Administration

Group	**Pretest**	**Treatment**	**Posttest 1**	**Posttest 2**
T1a	GTT	X	GTT	Test$T1$
T1b	Test$T1$	X	GTT	GTT
C1a	GTT		Test$C1$	
C1b	Test$C1$		GTT	
T2a	GTT	X	Test$T2$	GTT
T2b	Test$T2$	X	GTT	GTT
C2a	GTT		GTT	
C2b	Test$C2$		GTT	

Figure 4.1: Lesson 1, Question 1



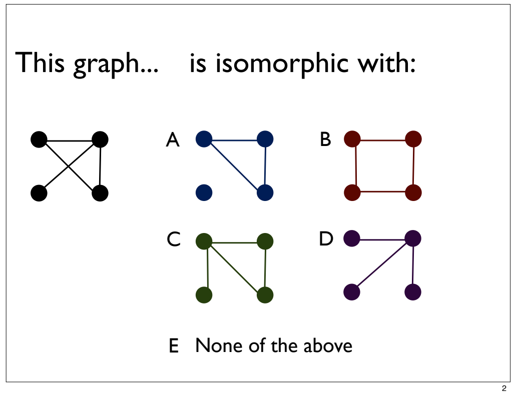
asked to vote whether each was correct or incorrect. During the iClicker activity, anonymous student responses to each question were shown in a pie chart, once all participants had responded. This student data was used to initiate a discussion about the graphs and address knowledge gaps. During this activity, students were highly engaged, and quite boisterous. In the T1 class, some students were mildly disruptive, and for some questions, students "voted" on invalid answers (see Table 4.2). Since all questions were true or false, only two of the five buttons on the iClickers were associated with intended responses. This problem was corrected in future iClicker activities by providing five possible responses to all questions. For a more detailed description of the lesson (see Appendix: A.7.1). For all tables of iClicker data, correct answers are indicated in **bold**.

Table 4.2: T1, T2: iClicker Activity 1 (Correct answers in **bold text)**

	T1				**T2**			
Question	**Yes**	**No**	**n/a**	**Correct**	**A**	**B**	**n/a**	**Correct**
1: Is this graph correct?	3	**19**	0	86.4%	1	**26**	1	92 9%
2: Is this graph correct?	**19**	3	0	86.4%	**25**	3	0	89.3%
3: Is this graph correct?	2	**20**	0	90.9%	9	**19**	0	67.9%
4: Is this graph correct?	5	**15**	2	68.2%	8	**18**	1	64.3%
5: Is this graph correct?	**15**	2	5	68.2%	**25**	2	1	89.3%
6: Is this graph correct?	**14**	1	6	63.6%	**23**	4	1	82.1%

For all iClicker questions, more than half of the participants were able to correctly identify both correct and incorrect graphs. Participant responses to Q1 were most correct, with 86.36% of T1, and 92.86% of T2 correctly identifying that the graph in Q1 was not drawn correctly (one edge is not terminated in a vertex). Similarly, 86.4%

Figure 4.2: Lesson 2: Question 1



of T1 and 89.3% of T2 identified that the graph in Q2 was correct (see Figure 4.1). In response to Q3, 90.90% of T1 and 67.9% of T2 were able to correctly identify that the graph shown was incorrect. Fewer students were able to identify that the graph in Q4 was incorrect, 68.2% of T1, and 64.3% of T2. More participants in T2 indicated that the graphs shown in Q5 and Q6 were correct (89.3% and 82.1% respectfully), than those in T1 (68.18% and 63.6% respectfully) (see Table 4.2).

4.3 Treatment: Lesson 2

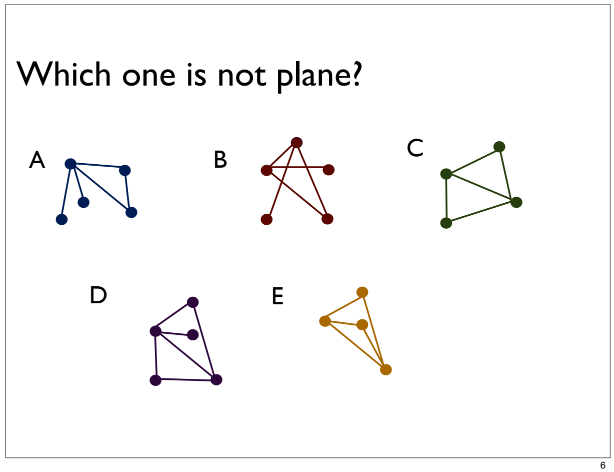
This lesson began with a review of previously covered material, followed by an introduction to two properties of graphs: isomorphism and planarity. Each half of the lesson included: instruction, group exercise, and individual practice sheet exercises. An iClicker activity and discussion occurred at the end of the lesson.

Questions 1-4 dealt with isomorphism. Two graphs are isomorphic if they can be redrawn (without modifying any connection information, or adding or removing any vertices) in such a way that they look the same (see Appendix A.1.2). In each case, students were shown a graph on the left, and asked to identify which of four graphs on the right (or none) was isomorphic with it (see Figure 4.2).

In both treatment groups at least 50% of participants were not able to correctly identify which of the graphs were isomorphic with the one given (see Tables 4.3, 4.4). For all questions but one more participants selected the correct answer than any single other answer. In T1, slightly more participants selected “None of the above” for their answer, rather than the correct answer.

In Question 5, participants were asked to identify the graph which was not plane.

Figure 4.3: Lesson 2: Question 5



A plane graph is one which is drawn with no crossing edges. Only 20% of participants in T1, and 44.4% of those in T2 indicated that graph B was plane (see Figure 4.3).

Question 6, asked which of five graphs was not planar. A planar graph is one which *can be drawn* with no crossing edges. In group T1, 60% of participants correctly identified graph C as non-planar. In group T2, only 11.11% of participants were able to identify graph C as non-planar. Most students, 65.4%, chose graph A, the only plane graph given, as non-planar (see Table 4.3 for details).

For a more detailed description of the lesson, see Appendix: A.7.3

4.4 Treatment: Lesson 3

This lesson began with a review of previously covered material, followed by an introduction of: trees, forests, cycles, connected components, and finally Minimum

Table 4.3: T1: iClicker activity 2 (Correct answers in **bold text)**

Question	**A**	**B**	**C**	**D**	**E**	**n/a**	**Correct**
1: Isomorphic	1	4	**5**	3	4	3	25.0%
2: Isomorphic	2	1	**8**	3	5	1	40.0%
3: Isomorphic	**10**	0	3	5	1	1	50.0%
4: Isomorphic	**4**	3	4	2	5	2	20.0%
5: Plane	0	**10**	3	2	2	3	50.0%
6: Non Planar	2	2	**12**	0	2	2	60.0%

Table 4.4: T2: iClicker activity 2 (Correct answers in **bold text)**

Question	**A**	**B**	**C**	**D**	**E**	**n/a**	**Correct**
1: Isomorphic	1	4	**8**	3	3	8	29.6%
2: Isomorphic	0	1	**13**	0	6	7	48.2%
3: Isomorphic	**7**	2	2	6	4	6	25.9%
4: Isomorphic	**12**	4	0	1	3	7	44.4%
5: Plane	0	**18**	2	1	0	6	66.7%
6: Non Planar	17	1	**3**	0	0	6	11.1%

Spanning Trees (MSTs) (see Appendix A.1). This lesson included discussion, group problem solving and individual practice sheet exercises. A tree is a connected graph with no cycles, like an evolutionary tree, for example. A forest is a graph made up of a set of one or more unconnected trees. A MST is a "cheapest" way to connect vertices in a graph into a tree (where the edges have some sort of cost) (see Appendix A.1). For a more detailed description of the lesson, see Appendix: A.7.4. As part of this lesson, participants were shown a short video about relational graphs [4].

4.5 Treatment: Lesson 4

After a review of the previous topics and terminology covered, this lesson introduced: paths, shortest paths, clusters and social networks. This lesson included an iClicker exercise to lead a discussion about clusters, and individual practice sheet exercises (for a more detailed description of the lesson, see Appendix: A.7.5). A path is a chain of vertices, connected by edges, which if one were to "walk" along the path no vertex would be visited more than once. A shortest path is a shortest path between a pair of given vertices. For the purpose of this lesson, a cluster is a set of connected vertices such that there is no path from any vertex in the cluster to a vertex outside

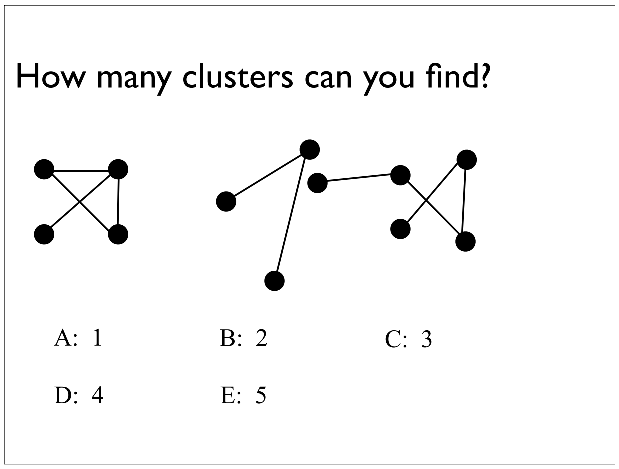
Table 4.5: T1: iClicker Activity 4 (Correct answers in **bold text)**

Question	**A**	**B**	**C**	**D**	**E**	**n/a**	**Correct**
1: Number of Clusters	0	1	**16**	1	0	1	84.2%
2: Number of Clusters	0	3	1	**11**	3	1	57.9%
3: Number of Clusters	0	2	10	**6**	0	1	31.6%
4: Biggest Cluster	1	**9**	1	0	7	1	47.4%
5: Smallest Cluster	**13**	2	3	0	0	1	68.4%

Table 4.6: T2: iClicker Activity 4 (Correct answers in **bold text)**

Question	**A**	**B**	**C**	**D**	**E**	**n/a**	**Correct**
1: Number of Clusters	1	1	**22**	0	1	1	84.6%
2: Number of Clusters	0	0	2	**22**	1	1	84.6%
3: Number of Clusters	0	0	19	**5**	5	2	19.2%
4: Biggest Cluster	0	**15**	5	4	0	2	57.6%
5: Smallest Cluster	**17**	2	5	0	0	2	65.3%

Figure 4.4: Lesson 4: Question 1



the cluster (see Appendix A.1).

The iClicker exercise for this lesson included 5 questions in which students were asked to identify cluster properties of graphs. Questions 1-3 asked students to identify the number of clusters in a given graph (see Figure 4.4 for an example). For questions Q1 and Q2 the majority (Q1: 84.2% from T1 and 84.6% from T2. Q2: 57.9% from T1, and 84.6% from T2) of participants in both groups were able to correctly indicate the number of clusters in the graph shown. More than half of the participants in T1 and T2 incorrectly indicated that the graph in Q3 had three clusters. See Tables 4.5, 4.6 for details.

Question 4 asked to identify the biggest cluster, and question 5 asked to identify the smallest cluster. In both groups, more students selected the correct number of clusters to both of these questions than any other single question.

4.6 Treatment: Lesson 5

The final lesson introduced two concepts: graph colouring and using graphs to solve scheduling problems. As with all other lessons, it included an introduction to topics and terminology covered in previous lessons. The lesson included a discussion about minimum colouring of graphs, and individual practice sheet exercises, as well as an iClicker activity on graph colouring. For a more detailed description of the lesson, see Appendix A.7.6.

Table 4.7: T1: iClicker Activity 5 (Correct answers in **bold text)**

Question	A	B	C	D	E	n/a	Correct
1: Minimum Colouring	0	**16**	2	3	1	0	72.7%
2: Minimum Colouring	1	1	**18**	0	0	2	81.8%
3: Minimum Colouring	1	1	**17**	2	0	1	77.3%
4: Minimum Colouring	1	2	**8**	7	4	1	36.4%
5: Minimum Colouring	1	**7**	9	1	2	2	31.8%
6: Minimum Colouring	1	0	3	**15**	0	3	68.2%

Table 4.8: T2: iClicker activity 5 (Correct answers in **bold text)**

Question	A	B	C	D	E	n/a	Correct
1: Minimum Colouring	0	**16**	3	4	1	0	66.7%
2: Minimum Colouring	2	1	**21**	0	0	0	87.5%
3: Minimum Colouring	1	2	**19**	2	0	0	79.2%
4: Minimum Colouring	0	1	**4**	16	3	0	16.7%
5: Minimum Colouring	0	**10**	6	4	4	0	41.7%
6: Minimum Colouring	1	0	1	**20**	0	2	83.3%

To "colour" a graph, vertices are coloured so that no two neighboring vertices (two neighboring vertices are connected by an edge) are coloured with the same colour. A minimum colouring is one which uses the least number of colours to colour the graph (see Appendix A.1). The majority of students in both groups were able to correctly identify the minimum number of colours needed to colour the graphs in questions Q1-3 and Q6. In group T1, slightly more (41.0% vs. 31.8%) participants selected answer C over the correct answer (B) for question Q5 (see Figures 4.5 and 4.6). In group T2, significantly more (66.7% vs. 16.7%) selected the incorrect (C) graph over the correct graph (C) for question Q5.

Figure 4.5: Lesson 5: Question 4

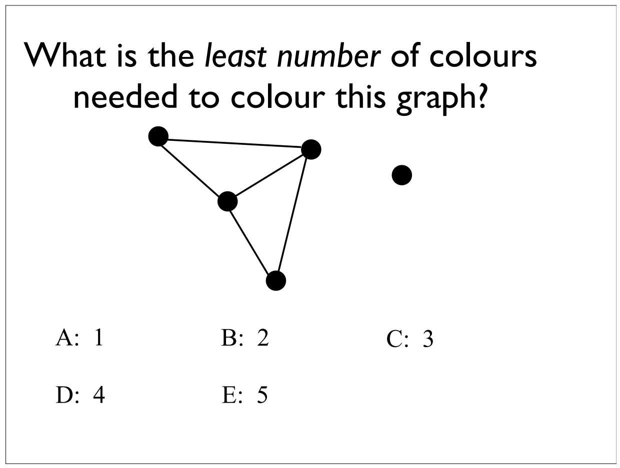
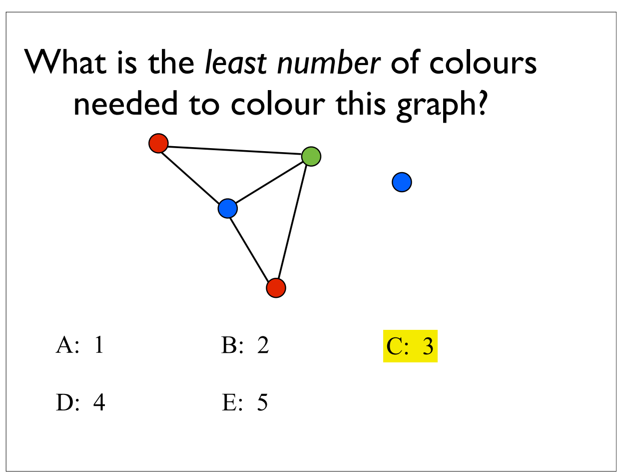


Figure 4.6: Lesson 5: Question 4 Solution



For this iClicker activity, after students had responded to each question, an example colouring of the given graph was presented to the class (see Figure 4.6 for an example).

4.7 Test Results

4.7.1 Mean Score

Participant scores on the GTT were summed across all four questions. Aggregate means of these scores were then calculated for two groups: all participants prior to any introduction to relational graph materials (GTT or instruction), and all participants after any introduction to relational graph materials (either GTT or instruction) (see Table 4.9 and Figure 4.7). Mean score across all participants was higher after being introduced to relational graphs either via the GTT or the intervention (or both).

Mean scores (and associated standard deviations) were also calculated on a class by class basis. Participants' mean scores in all groups increased from Pretest to their first Posttest (see Figures 4.8, 4.9, and Table 4.9). Both treatment groups' mean scores were higher on the second posttest relative to the pretest. However, group T1 saw a slight decrease in mean score from posttest 1 to posttest 2 (see Figures 4.8, 4.9, and Table 4.9). Standard deviations for all mean scores were relatively high, but decreased in all cases except group C1 (see Figure 4.8 and Table 4.9).

Table 4.9: Participant Mean Test Scores

		Mean	**St. Dev.**	**n**
Aggregate	Prior	20.0	6.1	79
	Post	22.1	65.3	50
T1	Pretest	20.0	6.0	8
	Posttest 1	24.0	3.5	18
	Posttest 2	22.0	3.3	10
T2	Pretest	17.0	8.1	12
	Posttest 1	20.4	6.9	11
	Posttest 2	24.0	6.3	24
C1	Pretest	19.0	2.7	4
	Posttest	18.8	3.6	5
C2	Pretest	17.9	4.7	13
	Posttest	20.2	4.5	22

Figure 4.7: Aggregate Mean Scores

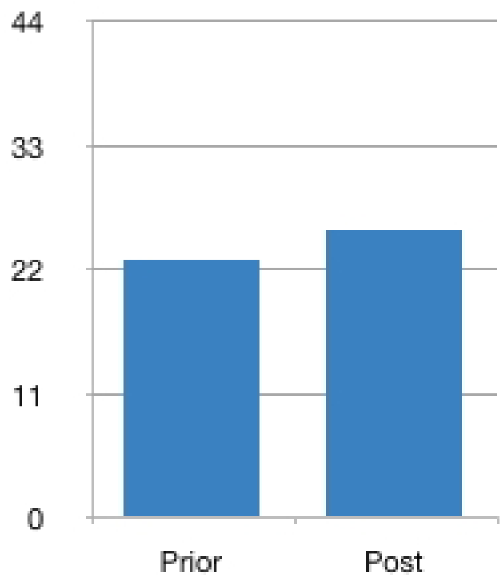


Figure 4.8: T1 & C1: Mean Scores

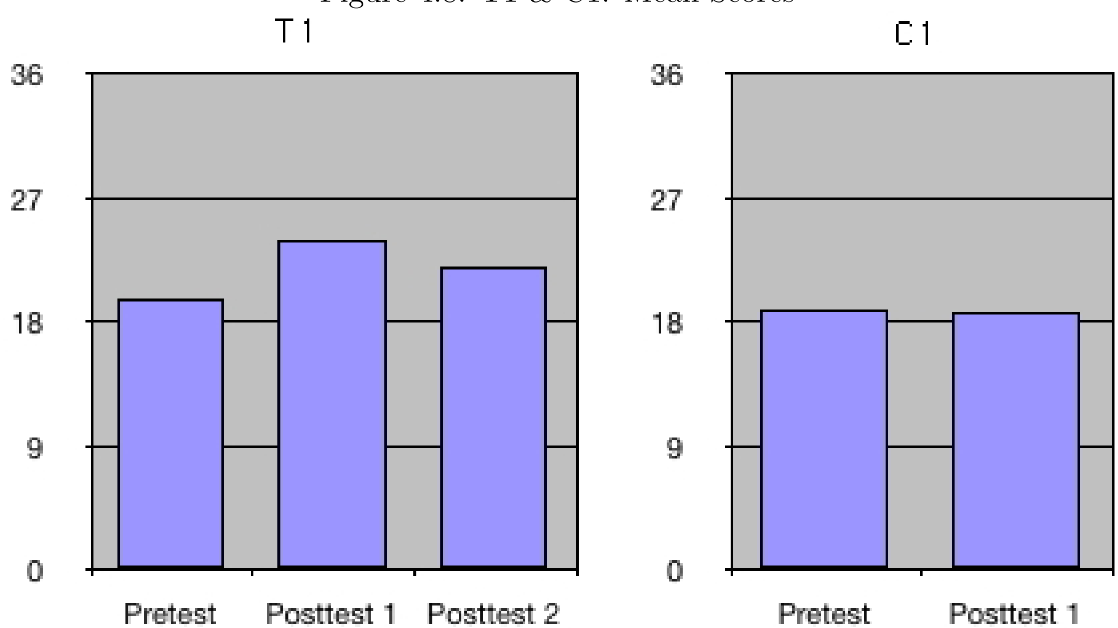
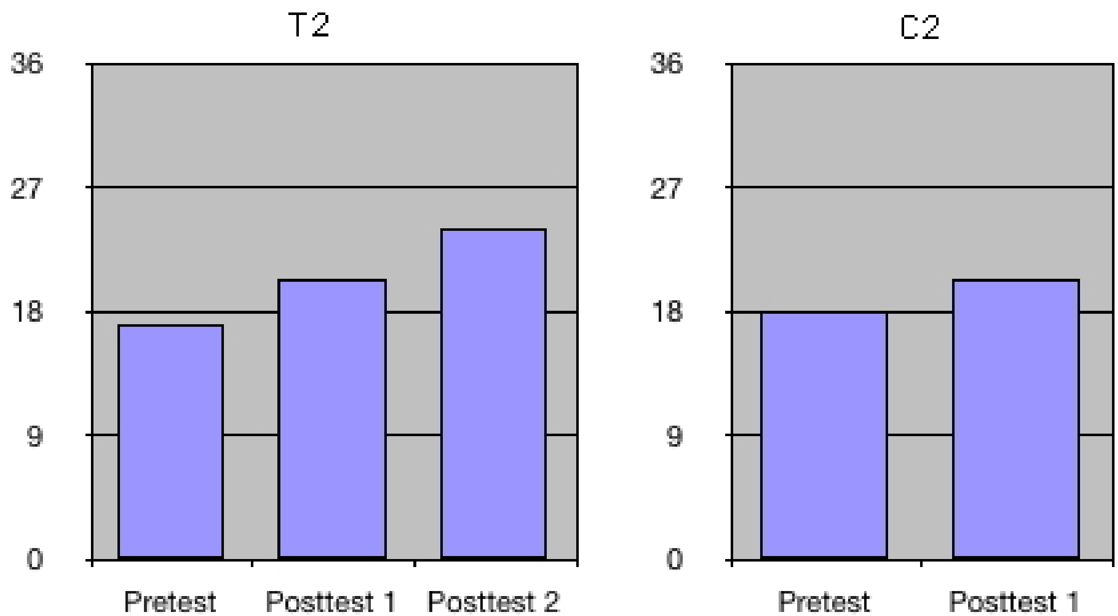


Figure 4.9: T2 & C2: Mean Scores



4.7.2 Draw a Graph

For each question on the GTT, participants were provided space in which they could work on the problem. It was hoped that they would use this space to "Draw a picture" to aid in their problem solving. A "Draw a Graph" (DAG) score was assigned to each question: 0 for no picture, 1 for a non-graph picture, and 2 for a graph. A picture was considered a graph if it included lines which connected entities (symbols, words, characters or images). In some cases students drew lines between words given in the problem text. This was treated as a graph.

For an example of a student drawing which was not considered a graph see Figure 4.16, and for an example of a drawing which was considered a graph see Figure 4.19.

Due to the small cell counts, the following results must be treated as exploratory in nature. Furthermore, since cell counts in most cases are less than five (5) for McNemar's test, significance levels (high or low) are suspect. This small cell count is inevitable due to the small number of participant responses available for the tests.

Note that no DAG data is presented for group C1 due to the lack of pair-wise pre-posttest data for this group.

Question 1

In treatment groups T1a and T2a, all participants drew a graph for Q1 when tested after the intervention. In T1a two participants who did not draw a graph on the pretest did draw one on the posttest, and in T2a six participants who drew no graph on the pretest did draw one on the posttest (see Table 4.10). No statistics were calculated for these crosstabulations, as the posttest data was constant. Therefore no significance value is known. In the control group C2a not all participants drew a graph on the posttest (see Table 4.10). All cell counts were below the minimum cell count (5) recommended for the test.

Question 2

When answering Q2, only one participant in group T1a drew a graph on the pretest, compared with four on the posttest. Of these four, three drew no picture at all on the pretest (see Table 4.11). In group T2a, none of the participants drew a graph on the pretest, and five drew a graph on the posttest (see Table 4.11). No statistics were calculated for this group, because pretest values were constant. In control group C2a,

Table 4.10: Q1 Draw a Graph

T1a			Posttest 1			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	0	0	2	2
		% Total	0.0	0.0	25.0	25.0
	Picture	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Graph	Count	0	0	6	6
		% Total	0.0	0.0	75.0	75.0
Total		Count	0	0	8	8
		% Total	0.0	0.0	100.0	100.0
T2a			**Posttest 2**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	0	0	3	3
		% Total	0.0	0.0	27.3	27.3
	Picture	Count	0	0	3	3
		% Total	0.0	0.0	27.3	27.3
	Graph	Count	0	0	5	5
		% Total	0.0	0.0	45.5	45.5
Total		Count	0	0	11	11
		% Total	0.0	0.0	100.0	100.0
C2a			**Posttest 1**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	0	2	2	4
		% Total	0.0	18.2	18.2	36.4
	Picture	Count	0	3	1	4
		% Total	0.0	27.3	9.1	36.4
	Graph	Count	0	0	3	3
		% Total	0.0	0.0	27.3	27.3
Total		Count	0	5	6	11
		% Total	0.0	45.5	54.5	100.0

no participants drew a graph on either the pretest or posttest (see Table 4.11). For all T1a, T2a, and C2a, all cell counts were below 5.

Table 4.11: Q2 Draw a Graph

T1a			**Posttest 1**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	3	0	3	6
		% Total	37.5	0.0	37.5	75.0
	Picture	Count	1	0	0	1
		% Total	12.5	0.0	0.0	12.5
	Graph	Count	0	0	1	1
		% Total	0.0	0.0	12.5	12.5
Total		Count	4	0	4	8
		% Total	50.0	0.0	50.0	100.0
T2a			**Posttest 2**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	6	0	5	11
		% Total	54.5	0.0	45.5	100.0
	Picture	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Graph	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	6	0	5	11
		% Total	54.5	0.0	45.5	100.0
C2a			**Posttest 2**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	11	0	0	11
		% Total	100.0	0.0	0.0	100.0
	Picture	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Graph	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	11	0	0	11
		% Total	100.0	0.0	0.0	100.0

Question 3

In group T1a, four participants drew a graph on the pretest, compared to six on the posttest, McNemar's of .500 (see Table 4.12). In group T2a, three participants who drew no picture on the pretest drew a graph on the posttest, and one student who drew a non-graph picture on the pretest, drew a graph on the posttest, (see Table 4.12). No participants in group C2 drew a graph on the posttest (see Table 4.12).

Table 4.12: Q3 Draw a Graph

T1a			Posttest 1			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	2	0	2	4
		% Total	25.0	0.0	25.0	50.0
	Picture	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Graph	Count	0	0	4	4
		% Total	0.0	0.0	50.0	50.0
Total		Count	2	0	6	8
		% Total	25.0	0.0	75.0	100.0
T2a			**Posttest 2**			
			No Picture	Picture	**Graph**	**Total**
Pretest	No Picture	Count	4	1	3	8
		% Total	40.0	10.0	30.0	80.0
	Picture	Count	0	1	1	2
		% Total	0.0	10.0	10.0	20.0
	Graph	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	4	2	4	10
		% Total	40.0	20.0	40.0	100.0
C2a			**Posttest 2**			
			No Picture	Picture	**Graph**	**Total**
Pretest	No Picture	Count	9	1	0	10
		% Total	81.8	9.1	0.0	90.0
	Picture	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Graph	Count	1	0	0	1
		% Total	9.1	0.0	0.0	9.1
Total		Count	10	1	0	11
		% Total	90.9	9.1	0.0	100.0

Question 4

There were no significant results in terms of the types of drawings students drew for Q4 (see Table A.1). For group T1a the McNemar test $(2, n = 8) = .333$, $p = .846$, and for all other groups no significance values were available.

DAG Sum

Table 4.13: DAGSUMCAT

T1a			Posttest 1		
			0	1	**Total**
Pretest	0	Count	2	3	5
		% Total	25.0	37.5	62.5
	1	Count	0	3	6
		% Total	0.0	37.5	37.5
Total		Count	2	6	8
		% Total	25.0	75.0	100.0
T2a			**Posttest 2**		
			0	1	**Total**
Pretest	0	Count	7	4	11
		% Total	63.6	36.4	100.0
	1	Count	0	0	4
		% Total	0.0	0.0	0.0
Total		Count	7	4	11
		% Total	63.6	36.4	100.0
C2a			**Posttest 1**		
			0	1	**Total**
Pretest	0	Count	9	0	9
		% Total	100.0	0.0	100.0
	1	Count	0	0	0
		% Total	0.0	0.0	0.0
Total		Count	9	0	9
		% Total	100.0	0.0	100.0

Participant DAG scores were summed (DAGSUM), and a categorical value (DAGSUMCAT) was created from these scores: DAGSUMCAT $= 0$, if DAGSUM ≤ 3, DAGSUMCAT $= 1$ otherwise ($0 \leq$ DAG < 2, therefore $0 \leq$ DAGSUM < 7). McNemar's test yielded the following results on these categorical scores. A DAGSUMCAT score of 0 indicates that the presence of at most one graph in all four of the student solutions. An increase of DAGSUMCAT score from 0 to 1, indicates that for at least one question a shift occurred (either from no drawing to a non-graph picture or a graph, or from a non-graph picture to a graph). A shift from DAGSUMCAT score from 1 to 0 indicates the inverse.

In both treatment groups, there is a shift from a DAGSUMCAT score of 0 to a score of 1, while in the control group there is no shift. In group T1, three participants who scored 0 on the pretest, scored 1 on the posttest. No participants' DAGSUMCAT score went from 1 to 0 (see Table 4.13). Similarly, in group T2, four participants who scored 0 on the pretest, scored 1 on the posttest, and no scores went from 1 to 0 from pretest to posttest. No statistics were computed for these tests, as the DAGSUMCAT value for the pretest was constant (all participants scored 0 (see Table 4.13) on the pretest). In group C2, all participants scored a DAGSUMCAT of 0 on both the pre- and posttests. No statistics were calculated, as the values were constant (see Table 4.13).

4.7.3 Graph Usage and Correctness

In order to test the hypothesis that the use of relational graphs to solve mathematical problems positively impacts students mathematics achievement, a crosstabulation of DAG score and categorical correctness score for each of the four questions was performed. A categorical score was created for each of the four GTT questions. Because of the ordinal nature of the data, the Tau measure of association was used to measure the association between the correctness of participants' responses, and graph usage.

Again, due to small cell counts high and low p-values are suspect, and may not be indicative of the significance of the results. As such, all results are treated as exploratory in nature.

Pretest Question 1

There may be a moderate association between drawing a graph and a high correctness score for group T1, $\tau(n = 8) = .655, p = 0.221$ (see Table 4.14). There may be a low association between drawing a graph and a high correctness score for group T2, $\tau(n = 11) = .259, p = 0.392$ (see Table 4.14). The high p-values indicates that we cannot claim significance. An association between drawing a graph and higher correctness scores was found for group C1, $\tau(n = 4) = .775, p = .083$ (see Table 4.14). No association between drawing a graph and higher correctness scores was found for group C2, $\tau(n = 13) = .000, p = 1.000$ (see Table A.2).

Table 4.14: T1 & T2: Pretest Q1 DAG Correctness

T1a			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	1	0	0	1
		% Total	12.5	0.0	0.0	12.5
	2.0	Count	1	0	6	7
		% Total	12.5	0.0	75.0	87.5
Total		Count	2	0	6	8
		% Total	25.0	0.0	75.0	100.0

T2a			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	1	0	0	1
		% Total	9.1	0.0	0.0	9.1
	1.0	Count	0	2	1	3
		% Total	0.0	18.2	9.1	27.3
	2.0	Count	2	1	4	7
		% Total	18.2	9.1	36.4	63.6
Total		Count	3	3	5	11
		% Total	27.3	27.3	45.5	100.0

Posttest Question 1

No association between drawing a graph and a high correctness score was found for group T1 or T2. No statistics were calculated, as DAG was constant (2.0) (see Table 4.15). A negative association between drawing a graph and a high correctness score was found for group C1, $\tau(n = 5) = -0.408, p = 0.232$ (see Table 4.15). A very weak association between drawing a graph and a high correctness score was found for group C2, $\tau(n = 22) = .113, p < 0.650$ (see Table 4.17), although the high p-value indicates that this may not be significant.

Table 4.15: T1 & T2: Posttest 1 Q1 DAG Correctness

T1			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	0	0	2	2
		% Total	0.0	0.0	12.5	12.5
	2.0	Count	0	0	14	14
		% Total	0.0	0.0	87.5	87.5
Total		Count	0	0	16	16
		% Total	0.0	0.0	100.0	100.0
T2			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	0	0	3	2
		% Total	0.0	0.0	37.5	37.5
	2.0	Count	0	0	5	5
		% Total	0.0	0.0	62.5	62.5
Total		Count	0	0	8	8
		% Total	0.0	0.0	100.0	100.0

Pretest Question 2

For all groups except T1, the DAG value for Q2 on the pretest was constant (no picture), therefore no meaningful associations can be interpreted from the data (see Table: A.3). For group T1, there appears to be a weak inverse association between not drawing a picture and correctness (see Table A.3).

Table 4.16: T1 & T2: Posttest 2 Q1 DAG Correctness

T2			DAG No Picture	DAG Picture	DAG Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	0	0	1	1
		% Total	0.0	0.0	5.3	5.3
	2.0	Count	0	0	18	18
		% Total	0.0	0.0	94.7	94.7
Total		Count	0	0	19	19
		% Total	0.0	0.0	100.0	100.0

T1			DAG No Picture	DAG Picture	DAG Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	0	0	2	2
		% Total	0.0	0.0	22.2	22.2
	2.0	Count	0	0	7	7
		% Total	0.0	0.0	77.8	77.8
Total		Count	0	0	9	9
		% Total	0.0	0.0	100.0	100.0

Table 4.17: C1 & C2: Posttest 1 Q1 DAG Correctness

C1			DAG No Picture	DAG Picture	DAG Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	0	0	1	1
		% Total	0.0	0.0	20.0	20.0
	2.0	Count	0	2	2	4
		% Total	0.0	40.0	40.0	80.0
Total		Count	0	2	3	5
		% Total	0.0	40.0	60.0	100.0

C2			DAG No Picture	DAG Picture	DAG Graph	**Total**
Correctness	0.0	Count	1	0	0	1
		% Total	4.5	0.0	0.0	4.5
	1.0	Count	0	2	3	5
		% Total	0.0	9.1	13.6	22.7
	2.0	Count	0	8	8	16
		% Total	0.0	36.4	36.4	72.7
Total		Count	1	10	11	22
		% Total	4.5	45.5	50.0	100.0

Posttest Question 2

There is no association between drawing a graph and correctness of student solutions to Q2 on the posttest in any groups (see Tables A.5, A.6).

Pretest Question 3

There is no association between drawing a picture and correctness scores for solutions to Q3 in all groups on the pretest (see Table A.7).

Posttest Question 3

There is no association between drawing a picture and correctness scores for solutions to Q3 in all groups on the posttest (see Tables A.8, A.9).

Pretest Question 4

There is no association between drawing a picture and correctness scores for solutions to Q4 in all groups on the pretest (see Tables A.10).

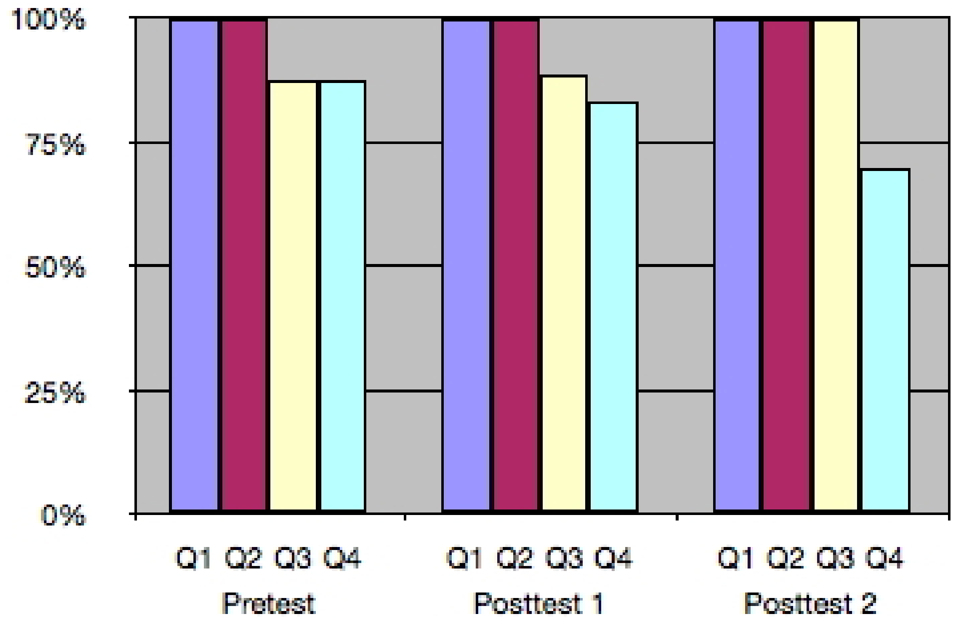
Posttest Question 4

There is no association between drawing a picture and correctness scores for solutions to Q4 in all groups on the posttest (see Tables A.11, A.12).

4.7.4 Engagement/Opportunity

In response to apparent discrepancies in participant engagement levels on questions on the GTT, a categorical engagement score was devised for each of the four questions. Students may have had varying levels of engagement on each of the questions. Or, an alternative explanation is that some students may have had less opportunity to engage with certain questions for a number of reasons, including lack of time or lack of comprehension. Participant GTT responses to each of the question were coded by the researcher: 0 for low engagement, 1 for high engagement. A response was scored a 0 if it appeared the student: didn't try at all (the page was not marked), expressed an inability ("I do not understand the question"), or gave up (as evidenced by a minor attempt at the problem). A response was scored 1 if the student either: answered the question, or seemingly put effort into attempting the question (as indicated by

Figure 4.10: T1: Engagement



notes, drawings, calculations, even if erased). The following figures show the count of engaged responses to each of the questions in the treatment and control groups.

In all groups, both Treatment and Control, participant engagement on Q3 and Q4 was lower than that on Q1 and Q2. In all groups on all tests except T2 Posttest 2 and C2 Pretest participant engagement on Q4 was lower than that on Q3 (see Figures 4.10, 4.11, 4.12, 4.13).

4.7.5 Student Representation of Social Networks

In response to patterns which emerged from some of the student responses, purposefully selected pair-wise participants' drawings of social networks are presented in the following section.

Some participant drawings of social networks (in response to Q1 of the GTT) changed following the intervention. For example, students changed from not drawing a graph (Figure 4.16) to drawing a graph (Figure 4.17), where other participants' representation of graphs evolved (improved) from before the intervention (Figures 4.14, 4.18) to after (Figures 4.15, 4.19). For example, participant 14 in T1 drew directional arrows between stick figures in the drawing for Q1 prior to the intervention, whereas the the same student drew non-directional edges between very simple vertices

Figure 4.11: T2: Engagement

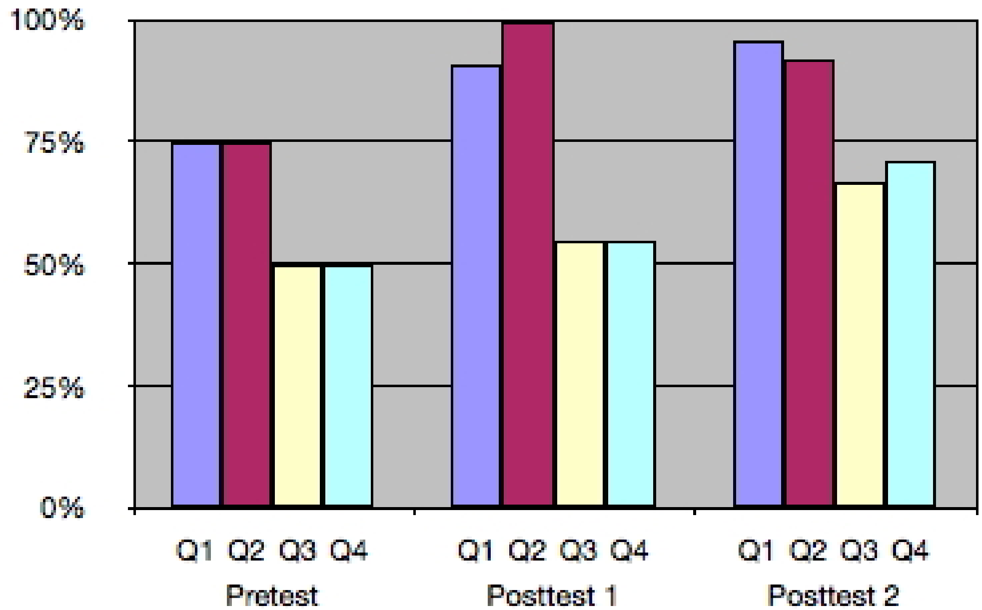


Figure 4.12: C1: Engagement

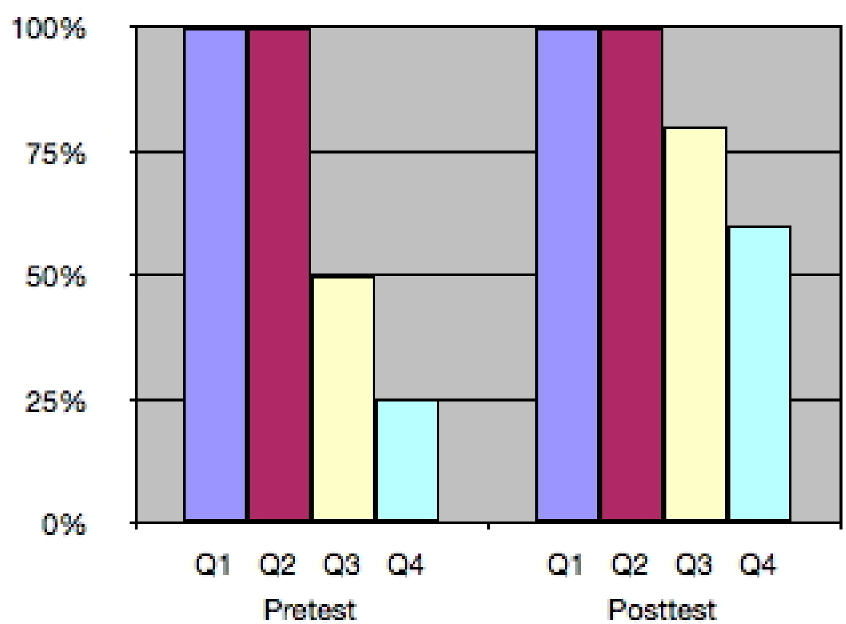
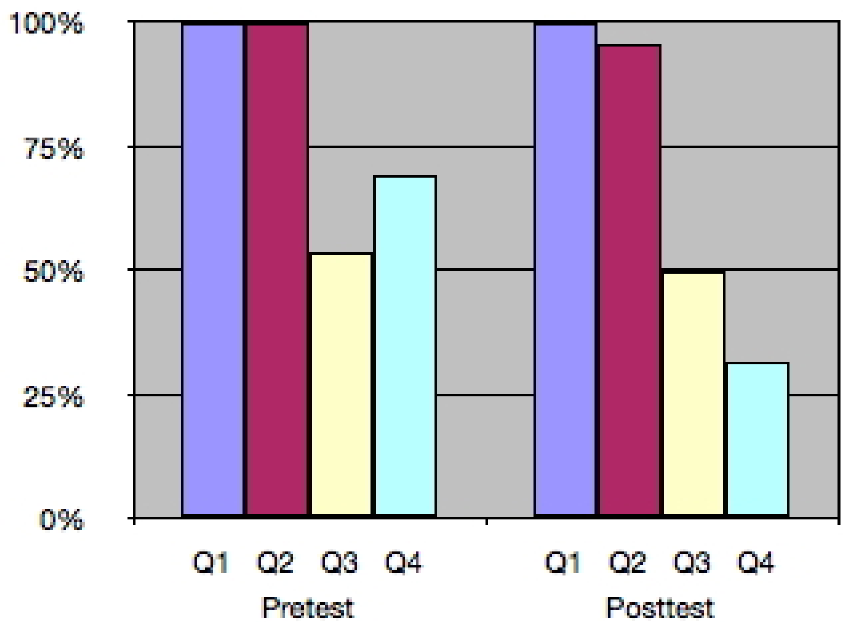


Figure 4.13: C2: Engagement



(filled in circles) on the GTT after the intervention. In group T1, participant 8's graph on the pretest GTT contained duplicates of all vertices, in contrast to their graph on the posttest, which contained no such duplications. Participant 19 in group T2 also drew stick figures in their graph on the pretest, and not on the posttest (see Figures 4.20, 4.21).

A number of participants' responses to Q1 included a high degree of detail, something which was not present in any of the responses to Q1 after the intervention. Vertices in students' graphs on the pretest included features such as: hair, eyes, bodies, hats and clothes (see Figures 4.22, 4.23). In no case were these features present on posttest responses of those participants who participated in the intervention. Some participants included words like *Friends* or *BF* (Best friends) in their drawings.

Figure 4.14: T1: Participant 08 Pretest Q1

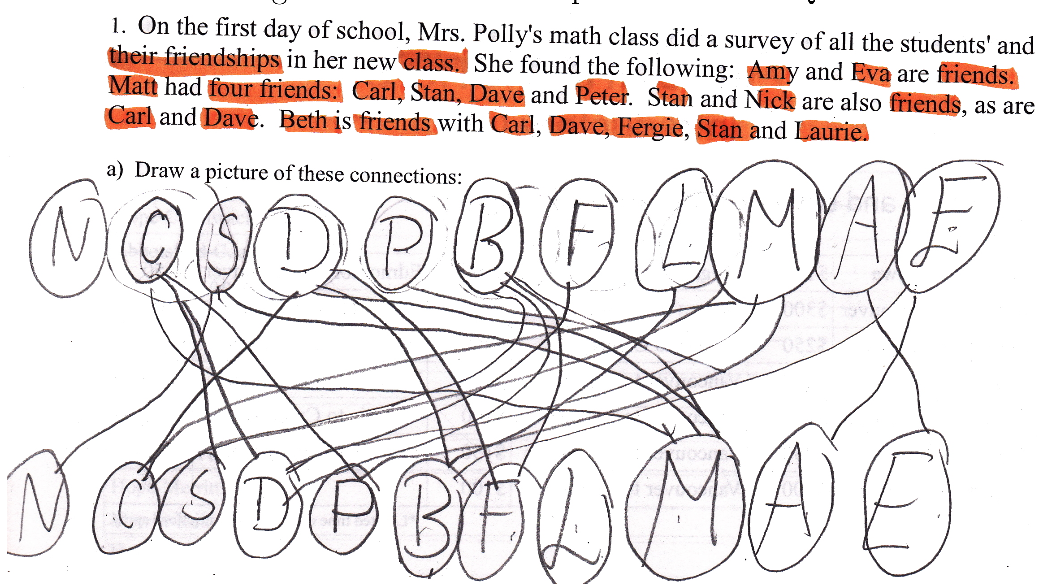


Figure 4.15: T1: Participant 08 Posttest Q1

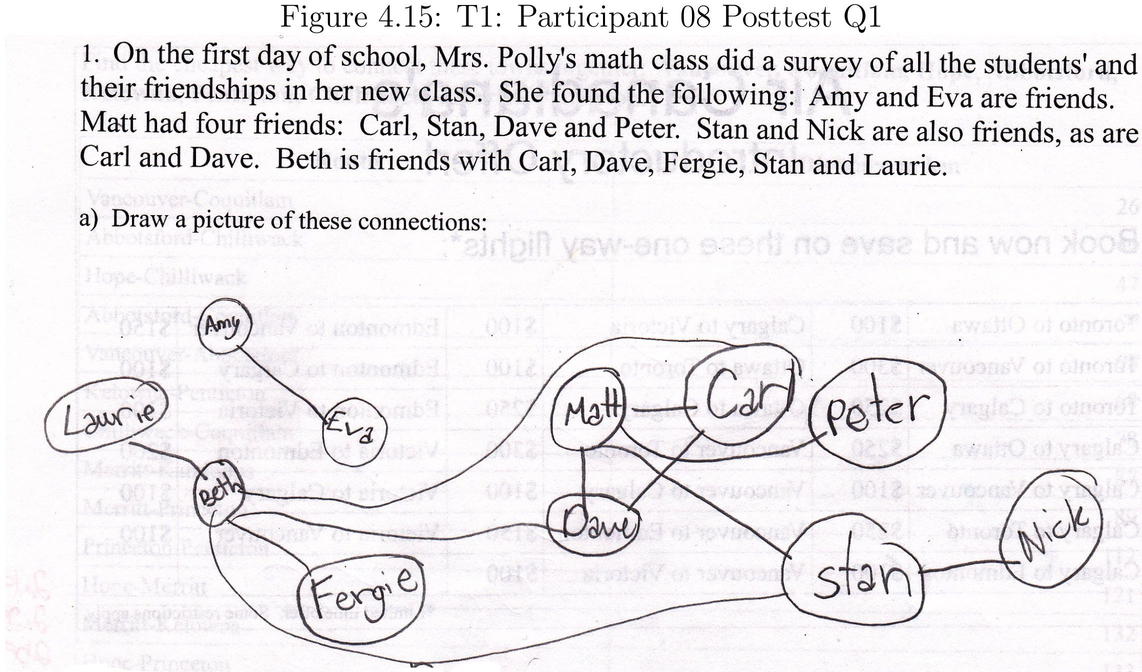


Figure 4.16: T2: Participant 09 Pretest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

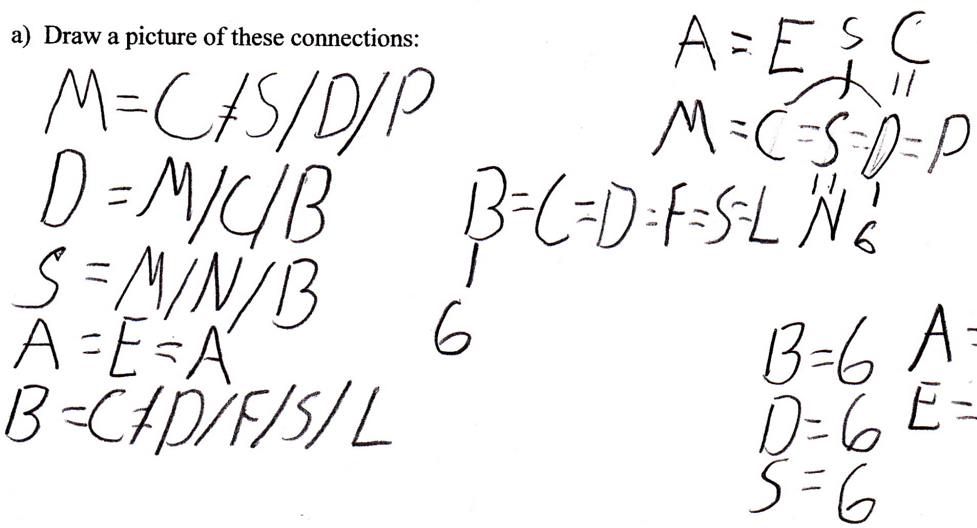


Figure 4.17: T2: Participant 09 Posttest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

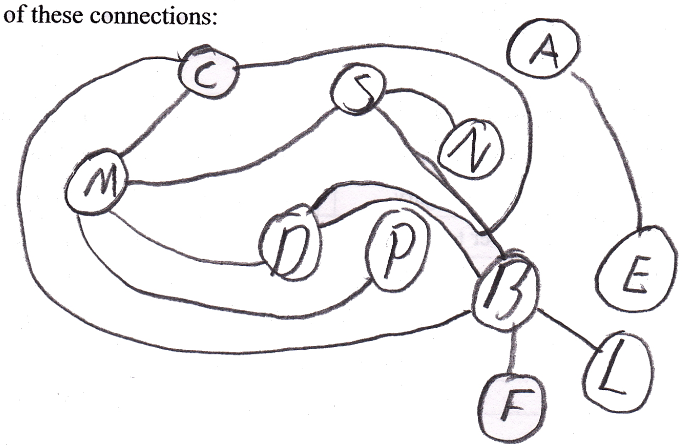


Figure 4.18: T1: Participant 14 Pretest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

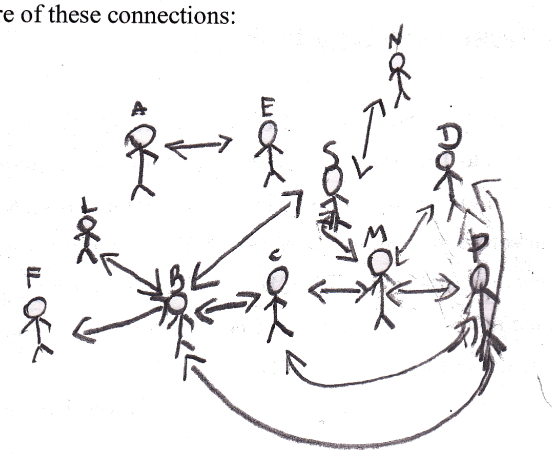


Figure 4.19: T1: Participant 14 Posttest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

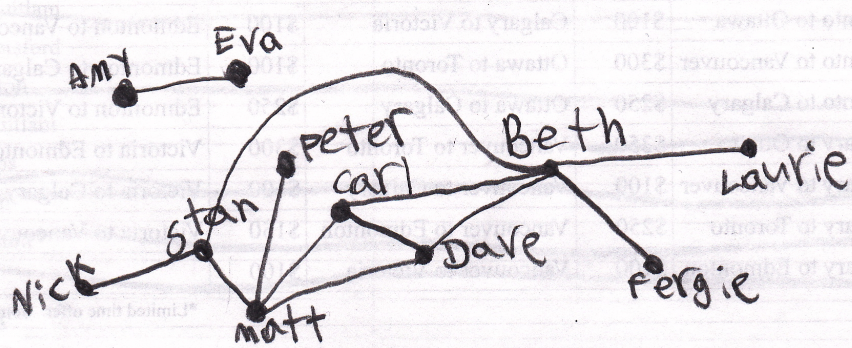


Figure 4.20: T2: Participant 19 Pretest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

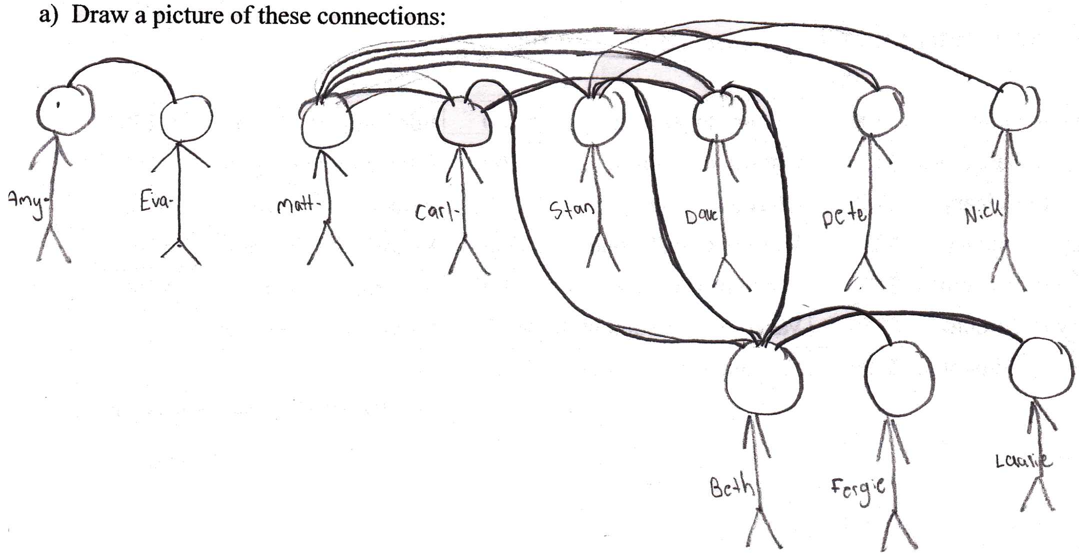


Figure 4.21: T2: Participant 19 Posttest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

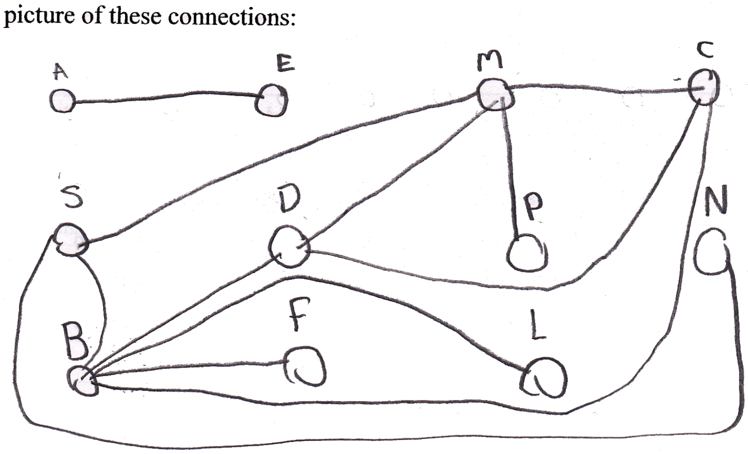


Figure 4.22: C1: Participant 10 Pretest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:

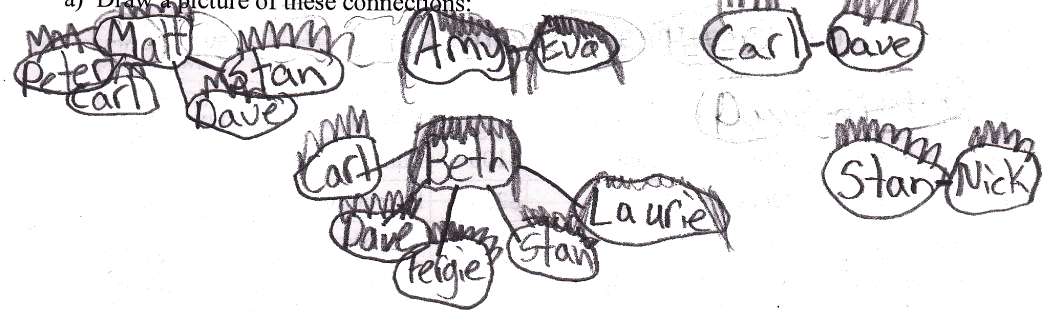
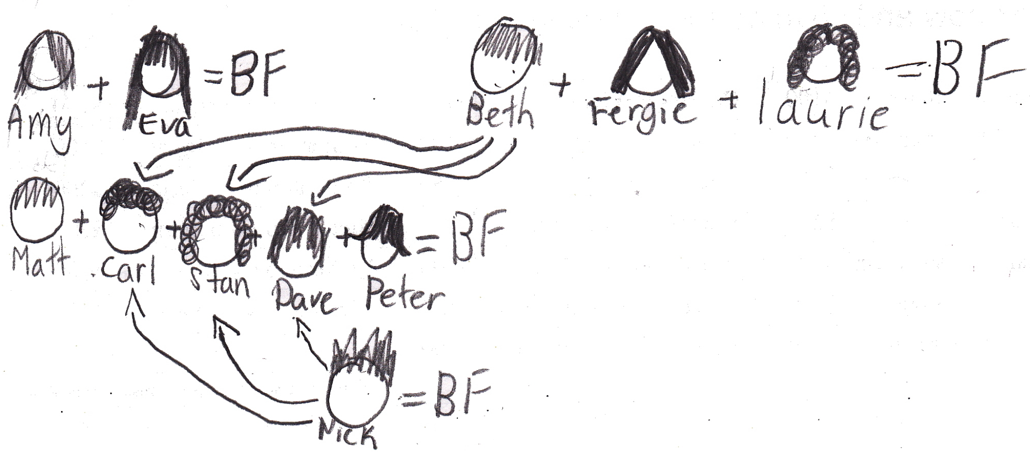


Figure 4.23: C1: Participant 25 Pretest Q1

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:



Chapter 5

Discussion

5.1 Ability to Learn Graph Theory

Research Hypothesis I: Grade Six students can learn to adopt relational graphs as one of many useful problem solving strategies.

5.1.1 First Steps

A fundamental first step in learning to adopt relational graphs to solve problems is to become familiar with, and proficient with the properties of, relational graphs. The following question is essentially a sub-question of the first research hypothesis.
Question: Can Grade Six students learn basic properties of relational graphs?

This first question is addressed by analyzing participant responses to iClicker questions which were part of the lessons.

iClicker Data

The following results, while not generalizable, indicate that students at this level are probably capable of learning graph theory with the process used in this study. Further study is warranted to verify if this is indeed the case.

In this section we look at participant understanding of graph theory based on the iClicker data collected during the lessons. There are limitations to this data, as students were able to discuss with peers and interact with each other during the process, and therefore their responses may have been influenced and may not be representative of their individual understandings. Furthermore, the student energy level was high during these iClicker activities, and some participants may have found the noise levels distracting. The fun nature of this data-gathering technique may have lead some participants to not take the responses as seriously as if they were engaged in a more formal assessment.

It appears that Grade Six students are able to grasp the basics of graph theory. By the end of the first lesson, more than half of the participants in both classes were able to identify both correct and incorrect relational graphs. I large proportion of participants were able to identify correct and incorrect graphs after the interventions (see Table 4.2).

If the iClicker responses are indicative of student comprehension, then they show that some students at this level are capable of understanding isomorphism and planarity (see Tables 4.3, 4.4). This comprehension appears weak. However, these are complex concepts, which were both taught in the same one hour lesson. The terms plane and planar are very similar, and this may be a source of some of the difficulty with these concepts. In particular, students may have incorrectly identified the plane graph as non-planar because they were focused on plane being different than planar. It remains to be determined if student comprehension of these concepts can be improved with further intervention, or if there is something fundamentally difficult about them for this stage of learner.

The concepts of clusters was comprehended by a large number of participants following instruction. Student responses to Question 3 were the least accurate, however, this was essentially a trick question, with two clusters presented in such that they overlapped. Most students in both classes incorrectly identified that there were three clusters rather than four, which is likely a result of accidentally misinterpreting these overlapping clusters as a single cluster. Considering the difficulty of this question, it is still remarkable that in both groups a significant number of participants selected the correct number of clusters (see Tables 4.5, 4.6). Student understanding of clusters may be connected to their level of engagement with the concept of social networking.

Finally, student responses to questions regarding the minimum number of colours needed to colour a graph were positive for the most part. In both classes the majority of students were capable of determining the minimum number of colours needed to colour the given graphs in four of six cases (see Tables 4.7, 4.8).

In future research, it would be of interest to determine if students at this level can better comprehend isomorphism and planarity. Because of the complexity of these concepts, it may be more appropriate to spend more time on these topics, and to present them as independent lessons. Perhaps creating a connection between isomorphism and planarity onto concepts which are relevant to participants would increase student understanding, as this appears to have had a positive impact on graph theory concepts like clustering.

5.1.2 GTT Data

The nature of the GTT data (small cell counts, missing control test data) implies that analysis of student test scores and responses is exploratory in nature. However, even though we cannot claim significance, we can still extract meaningful trends and patterns which can inform, and lead to, further study to investigate young learners' abilities to learn to adopt the use of graphs to solve problems.

A measure of students' ability to learn to apply relational graphs to solve problems is a shift in students' problem solving strategies from those which do not include graphs to those which do, following graph theory instruction. In Questions 1-3 there is a contrast in the trends of treatment and control group participant choice to draw a graph following the intervention, which seems to indicate that graph theory instruction impacts students' selections of appropriate problem solving strategies. While not necessarily statistically significant, it is still noteworthy.

In participants' responses to Question 1 of the GTT there is a trend in the treatment groups from not drawing a graph, prior to graph theory instruction, to drawing a graph following graph theory instruction (see Table 4.10). All post-intervention treatment group responses to Question 1 included a graph. In comparison, nearly half (5 of 11) control group solutions (C2) on the posttest did not include a graph (see Table 4.10). Although not statistically significant, this trend still indicates that students at this level may be capable of learning graph theory, and apply it when solving problems.

A similar, if slightly weaker, trend in students' use of graphs on Question 2 and 3 further supports this theory (see Tables 4.11, 4.12). In both treatment groups, there is a shift to students including graphs following graph theory instruction. Again, this trend is not seen in the control group (C2) where all participants included no graph (or picture) on either the pre- or post tests.

This trend is not as apparent on Question 4 (see Table A.1), where a large proportion of participants in all groups failed to include a graph or even a drawing in their solutions. However, considering the lower engagement/opportunity levels on this question, it is possible that the lack of drawings/graphs is a result of participants running out of time or becoming frustrated on this, final, question (see Figures 4.10, 4.11, 4.12, 4.13). Another possible reason for a lower association between the DAG score and graph theory instruction is the different level of abstraction in this problem. When using a graph to solve this problem, the students given in the problem can be used to generate the vertices. However, the edges must be generated from the implied conflicts in the students' schedules. This abstraction may have been too complex for students of this level to work with, and may have been discouraging to participants.

DAGSUMCAT scores support this theory as well. In both treatment groups there is an increase in the number of participants who scored a 1 on the posttest as compared to the pretest, in contrast to the control group in which all participants' DAGSUMCAT score remained at 0 (see Table 4.13). While these results may not be significant, they still hint at the treatment groups' ability to adopt some sort of drawing strategy in their problem solving. Since the control group shows no such trend, the introduction of graph theory in Math class may be responsible for this adoption of a new problem solving strategy on these problems.

5.1.3 Summary

The Grade Six students in this study appear capable of learning graph theory, and further some were even able to grasp complex concepts like isomorphism and planarity. Further, participants of this study were, for the most part, able to identify minimum colourings of graphs, as well as features like clusters. These are promising results, as these are first steps in learning to adopt relational graphs as problem solving tools. The results of the study also seem to support the first research hypothesis. The trends of student adoption of relational graphs in their solutions to Questions 1-3 on the GTT indicate that Grade Six students may indeed be able to learn to adopt relational graphs as one of many useful problem solving strategies. Further study is needed to verify these results, and to determine if they are generalizable.

5.2 Impact of Applying Graph Theory

Research Hypothesis II: The use of relational graphs to solve mathematical problems positively impacts students mathematics achievement.

5.2.1 GTT DAG Score and Correctness

There appears to be some positive association between the use of relational graphs and mathematics achievement as measured by the Tau measure of association between DAG scores and the correctness of student solutions. Again because of the small cell counts, we cannot claim statistical significance, however we can still infer some meaning from trends in this data.

In Question 1 there is a moderate association, in both T1 and C1, between students' DAG score and the correctness of solutions on the pretest (see Tables 4.15, 4.17). A similar, but somewhat weaker, association is apparent in group T2's DAG and correctness scores on the same test. This trend isn't apparent on the posttests. However, this may be due to the shift in the treatment groups to drawing graphs on the posttest. As a result, it is not possible to infer an association between these two variables. This could be remedied in future studies with a larger sample, which would likely result in a better distribution of student responses.

Data from Questions 2-4 don't seem to support this hypothesis, with no association between participant DAG score and the correctness of their solutions. For Question 2, the easiness of the problem may have partly responsible for the lack of graphs in student solutions. The difficulty of the problem may not have warranted the use of a graph. For Questions 3-4 it may be a result of the reduced engagement/opportunity level. It may also be that students were less likely to draw a picture (and subsequently less likely to draw a graph) because the questions did not explicitly ask for a drawing as part of the solution (unlike Question 1).

In future, to better investigate the association between students' use of relational graphs and achievement, studies could benefit from larger sample sizes. Further refinement of the GTT, and improvement of the questions may be needed. In order to determine if engagement/opportunity levels were responsible in the lack of trend on Questions 2-4, and further to refine understanding on the source of these lower engagement/opportunity levels, it would be wise to use multiple versions of the test with differing order of the questions.

5.2.2 Mean Scores

Another possible measure of the impact of graph theory instruction on student achievement, is the relative difference between the mean test scores of participants who either did, or did not, receive graph theory instruction. Posttest mean scores of both treatment groups are higher than those of the control groups (see Table 4.9). This difference could indicate that graph theory instruction equipped treatment group participants with new problem solving strategies which in turn allowed them to perform better on the posttests.

There are a other possible explanations for this difference as well. Participants in the control groups may have been less motivated on the posttests than those participants who were engaged in graph theory instruction. This reduced engagement could have contributed to lower performance. Further investigation is needed to determine if there is indeed a correlation between graph theory instruction, and test score on this type of test.

5.2.3 Summary

The use of relational graphs to solve mathematical problems seems to have had a weak positive impact on the mathematics achievement of the Grade Six students in this study. The strongest association between the use of relational graphs and achievement was seen in student responses to Question 1. However the statistical significance of this association is not known. Additional study of this association is needed. Participants of this study who received graph theory instruction did show a relative improvement (as compared to both Pretest and control group scores) in overall achievement. This may further support the second research hypothesis, however further study is needed to determine the significance and generalizability of these results.

5.3 Engagement/Opportunity

Trends in engagement/opportunity highlight possible issues with the GTT.

The general downward trend in engagement/opportunity from Question 1/2 to Question 3 and Question 4 (see Figures 4.10, 4.11, 4.12, and 4.13) may indicate that the test was too long or taxing for the participants. This could be confirmed in future studies by having multiple versions of the test which include only a subset of the questions. In order for this to work, however, a larger sample size would be needed to ensure large enough cell counts for analysis. Another option would be to administer the test question by question, thereby ensuring that participants do not miss one question because a disproportionate amount of time was spent on another.

These suggested improvements would increase the complexity of test administration and would likely require that the test were administered by the researcher or trained administrator.

It is also possible the trend is a result of the relative difficulty of Questions 3 and 4 (as compared to Questions 1 and 2). This could be tested by the aforementioned strategy of multiple tests with subsets of questions, or alternatively by offering different versions of the GTT with the questions in various orders. This might help in determining if the source of the reduced engagement/opportunity on latter questions was a result of prior questions.

Ultimately, until the source of this trend in engagement/opportunity is better understood, some results and analysis of the student responses to Questions 3 and 4, in particular, are muddied. Further piloting and refining of this test are needed before we can begin to understand the significance of patterns and trends in student responses.

5.4 Student Representation of Social Networks

A surprising outcome of the student responses on the GTT is the richness of student representations of social networks in Question 1. Prior to any introduction to relational graphs (either instruction or the GTT) student drawings of the connections between the friends given in the problem were much more rich in detail, including features such as: faces, hair, hats, glasses and bodies. It appears as though participants devoted a greater amount of time and energy to drawing these connections. There are a number of possible reasons for this.

Question 1 was the only question which explicitly asked students to "draw a picture", and so it is possible that students were less likely to choose not to draw a picture in response to Questions 2-4 because they didn't think it was required. However, it is also possible that this higher incidence of graph drawing, and greater level of detail in drawings (both graph and non-graph) is indicative of a stronger connection between the problem given and the participants' experiences. Friendships and connections are likely a more *real* concept to Grade Six students than concepts like plane travel or communications networks. Because of this, students may have been more able to relate to the problem. The use of terms like *BF* (Best Friends) in their solutions, something which doesn't seem to provide any extra information in terms of solving the problem, (see Figure 4.23) supports this hypothesis.

The trend of reduced detail, and simplification of solutions, on posttest graph representations also points to student adoption of abstraction following graph theory instruction. This is a fundamental feature of graphs. They provide a means of distilling a complex problem down to the desired relationships. Students in this study appear to have grasped this concept, as indicated by their choice to discard superfluous information when representing the connections in this question.

However, because the student solutions presented here were hand selected purposefully to highlight these features and level of detail, it is important to note that these results are not necessarily generalizable. In order to gain a better understanding of the relevance of this type of question, further analysis of the data gathered in this study is warranted. As well, future studies, designed specifically to evaluate student representations of social networks, can be designed to investigate what might be motivating the apparent student engagement on this questions.

A negative implication of this seemingly high level of engagement in Question 1, is that this may be the source of lower engagement in subsequent question responses. Students may have spent more time on Q1, and as a result ran out of time, focus or enthusiasm for later questions. For future administration of a test of this type, it may be wise to provide alternate versions of the test in which the questions are presented in different orders. This would allow for a differentiation between loss of engagement on questions because of time/energy spent on earlier questions, and problems with the questions themselves.

The potential implications of this are important. If this concept, that of connections and friendships, is more relevant for students of this age, then it should be possible to leverage it to develop meaningful activities and lessons, which can engage young learners in computer science topics.

5.5 Limitations Due to Testing Procedures and Participation Rates

As a result of deviations from the intended administration of the GTT, valuable information from this study may have been lost. In particular, the lack of Posttest 2 data in both control groups, combined with the misadministration of tests by one of the control group teachers, resulted in no pretest posttest pair-wise GTT data for one control group. However, some descriptive data in the form of distribution and mean is still salvageable. In addition the misadministration of tests by one of the treatment group teachers resulted in no pretest posttest pair-wise GTT data for that treatment group. This, fortunately, is offset by the fact that this treatment group completed the Posttest 2, and so pair-wise data is available for most participants in this group.

While teacher and student participation in the treatment groups was excellent, there was a lower level of participation (in terms of the number of returned consent forms, engagement/opportunity levels on GTT, and teacher communication) with both the control groups. This is not surprising, as the teachers and students in the treatment groups benefit more directly from participation in this type of study (new activities in the classroom, direct involvement with the researcher), where control groups gain much less in the short term by participating. This type of issue might be resolved in future studies by simultaneously running parallel studies where classes act as control for one study and treatment for another.

The end result is that the data collected from this study cannot be analyzed in the manner originally intended. Ultimately, the small group sizes in this study (both treatment and control) limit the generalizability of the results, and preclude most statistical analysis. Instead, the study can be treated as a pilot study, which still provides a great deal of insight into directions for future research. The source of these problems could be avoided in the future by either ensuring that all tests are administered by the researcher, or by trained assistants. Refinement of the instrument is also needed to clarify future results. Future studies would benefit from larger sample sizes, which implies the need for broader recruitment, better testing procedures, and/or higher participation rates.

5.6 Conclusion

This study was essentially a pilot study, and the results of the study are neither statistically significant nor generalizable the trends and patterns which have emerge from the data collected are promising. They indicate that further study of student understanding of, and ability to adopt, relational graphs is warranted. In addition, they also support further investigation of the association between the use of graphs and student achievement.

Chapter 6

Conclusions

Computer science is relatively young, as far as scientific disciplines go. It follows that computer science education research is still in its early stages of development. This is especially true when we consider computer science education research as it relates to elementary curriculum. In the late 1960's through the 1980's, computer science education was heralded to have great potential to support elementary learning. However, as evidenced by the lack of computer science present in elementary curricula, it appears that this expectation was either false or not realized.

Today, with the rapid adoption of computing devices worldwide, it is more important than ever to investigate what computer science topics can and should be taught in elementary classrooms. In light of previous efforts, it may be important to shift focus from programming and technology-based computer science teaching practices, to those which may be more suitable for a traditional classroom. Because of the diversity of computer science topics, there are many ways in which computer science can be taught. It need not be limited to programming instruction.

In this study we investigate the impact of unplugged computer science education on Grade Six students in a regular classroom setting. The study was designed as a quasi-experimental research study with control and treatment groups completing a pretest and two posttests. As a result of alterations to the intended research design, we are not able to claim statistical significance in the analysis of the data. However, we can still extract much meaningful information from the trends and patterns which emerged from student responses.

Grade Six students appear capable of not only learning graph theory, but applying it to solve problems. The use of relational graphs appears to positively impact student performance on at least some types of problem solving activity. An exciting, and unexpected result was the apparent engagement level of students, in particular on a question dealing with social networking. It appears that graph theory lends itself naturally to social network problems for this population. In all groups, both control and treatment, half of participants drew a graph when asked to draw a picture of connections between fictional friends on the posttest. Participants who took part in the intervention were all able to apply graph theory in working on this problem, with an apparent positive correlation with the correctness of their solution.

While this study didn't proceed as I had envisioned, the end result is that I learned a lot more about developing and deploying this type of research than I had imagined. I am buoyed by the trends and patterns which emerged from the data collected in this study. Further study of students' understanding of, and ability to adopt, graph theory are supported, not only at this grade level, but at earlier and later grade levels. The preliminary findings of this study should be verified with a much larger study so that we can begin to gain a more generalizable understanding of the impact of graph theory instruction on elementary achievement. Further, the detailed representation of social networks by participants in this study point to a need to further investigate how individuals at this age and stage of development perceive and represent friendships and connections.

From here I look forward, to take what I have learned from the design and implementation of this study and apply it to further research in this, and other parallel areas. I hope that this work and future endeavors can serve to further the understanding of the impact of computer science education on elementary students' achievement.

Appendix A

Appendices

A.1 Graph Theory

In computer science, a graph is a representation of relationships between entities. For this study, the term *relational graph* was used, rather than *graph*, in order to differentiate between those graphs students would normally refer to in their math class. This is consistent with the use of specific graph names such as: bar graph, line graph. The terms graph, and relational graph are interchangeable.

A graph consists of a set of vertices and a set of edges. Edges connect vertices, and represent a connection or relationship between vertices. Graphs can be represented in a number of ways, depending on the application, but for the purposes of this study, they are represented as a two-dimensional drawing, with vertices typically represented by circles or dots, and edges drawn as lines connecting vertices. However, note that various shapes can be used to denote vertices. Edges can be drawn as straight or curved lines with no impact on the meaning of the edge. Edges can be weighted, either arbitrarily, or based on their euclidean length. This value is often referred to as the cost of the edge. Vertices can be labeled and/or coloured.

Graphs can be directed or undirected. In directed graphs, the direction of an edge is shown with an arrow head. In undirected graphs, edges are represented simply as lines. For the purpose of this study, we consider only undirected graphs.

The following is a set of definitions which were touched on during the study.

A *loop* is an edge that connects a vertex to itself.

A *plane* graph is one which is drawn so that no edges cross or intersect.

A *planar* graph is one which can be drawn so that no edges cross.

Two graphs are said to be *isomorphic*, if they can be redrawn, without altering any of the connections between vertices, so that they are the same. More formally, two graphs G and H are isomorphic if there is a one-to-one correspondence between the graphs such that two vertices are connected in G if and only if their corresponding vertices in H are connected.

A *path* is a sequence of vertices $\{v_1, v_2, ..., v_n\}$,$n > 1$, and a set of edges $\{e_1, ...e_{n-1}\}$ such that edge e_i connects vertices v_i and v_{i+1}.

A *strongly connected component* of a graph is a subset of vertices which are connected by edges such that there is a path from one vertex in the subset to any other vertex in the subset. For the purposes of this study, we call this a *connected component*, or a *cluster*.

Two connected components $C1$ and $C2$ are said to be *unconnected* if there exists no path between any vertex in $C1$ and any vertex in $C2$. A graph is said to be connected if there is a connected component which contains all vertices of the graph.

A *cycle* is a path which starts and ends at the same vertex, and visits no vertex or edge more than once (except the start/end vertex).

A *tree* is a graph which is connected, and contains no cycles.

A *forest* is a graph which contains no cycles, but does not need to be connected.

A *minimum spanning tree* of a graph G is a cheapest tree on G. The cost of the tree is the sum of all edge-weights in the tree.

A *minimum colouring* of a graph is the minimum number of colours required to colour all the vertices of a graph such that no vertex is connected, by an edge, to a vertex of the same colour.

A.1.1 Seven Bridges of Königsberg Solution

The question of whether or not there exists a path which traverses all the bridges is the same as asking if there is a path on the graph given (see Figure 1.2) which traverses each edge exactly once. In general this is called an *Euler Path*.

The Seven Bridges of Königsberg graph does not have an Euler Path, so the answer is: No.

Euler proved that in order for a graph to have an Euler Path, there must be exactly zero or two vertices with odd degree. The degree of a vertex is the number of edges which are connected to that vertex. The Seven Bridges of Königsberg graph has four vertices with odd degree.

A.1.2 Rules of Correctness for Relational Graphs

1. an Edge must start and finish at a Vertex;
2. edges can be drawn as straight or curved lines;
3. a Vertex may be connected to another Vertex by an Edge, but doesn't have to be;
4. vertices can be labelled (or named), as long as there are no vertices with the same name;
5. and, if vertices in the graph are labelled, all vertices in the graph need to be labelled.

A.2 CSTA Grade-level Breakdown for Grades Six-Eight

Learning outcomes 1-10 suggested for grades 6-8 in the Computer Science Teacher's Association (CSTA) A Model Curriculum for K-12 Computer Science publication [11].

1. Apply strategies for identifying and solving routine hardware and software problems that occur during everyday use.
2. Demonstrate knowledge of current changes in information technologies and the effects those changes have on the workplace and society.
3. Exhibit legal and ethical behaviors when using information and technology and discuss consequences of misuse.
4. Use content-specific tools, software, and simulations (e.g., environmental probes, graphing calculators, exploratory environments, Web tools) to support learning and research.
5. Apply productivity/multimedia tools and peripherals to support personal productivity, group collaboration, and learning throughout the curriculum.

6. Design, develop, publish, and present products (e.g., Web pages, videotapes, using technology resources that demonstrate and communicate curriculum concepts to audiences inside and outside the classroom.

7. Collaborate with peers, experts, and others using telecommunications tools to investigate educational problems, issues, and information, and to develop solutions for audiences inside and outside the classroom.

8. Select appropriate tools and technology resources to accomplish a variety of tasks and solve problems.

9. Demonstrate an understanding of concepts underlying hardware, software, algorithms, and their practical applications.

10. Discover and evaluate the accuracy, relevance, appropriateness, conprehensiveness, and bias of electronic information sources concerning real-world problems.

A.3 Proposal

Computer Science Constructs for Problem Solving by Grade Six Students

Sarah Carruthers, University of Victoria

Purpose of the Study

The purpose of this study is to determine whether or not grade six students can adopt the use of theoretical computer science constructs like relational graphs to successfully problem solve.

This study is built upon the following previous studies: Higgens (1997) [5], Hoyles and Kuchemann (2003) [6], Montegue and Applegate (2000) [8] and Niman (1975) [9].

A preliminary literature search indicates that there are very few educational research studies of middle school (Grade 6 to 8) students' use of computer science problem solving techniques to solve grade-appropriate problems. This study will determine whether or not Middle School, more specifically grade six, students can learn to use computer science problem solving tools and techniques like relational graphs in their problem solving strategies.

Research Questions and Hypothesis

Can grade six math students use computer science constructs like relational graphs to effectively solve grade-appropriate mathematics problems? How does learning to use computer science constructs like relational graphs to solve mathematics problems affect grade six students' mathematics achievement, attitude and problem solving strategies? How does learning to use computer science constructs to solve mathematics problems affect grade six students' attitude towards theoretical computer science?

The research hypothesis is as follows: Grade Six students can learn to adopt relational graphs as one of many useful problem-solving strategies.

Literature Search

The following descriptors will be used in the initial literature search to identify relevant studies: *Middle School, Computer Science, Computer Science Educational Research, Graph Theory, Draw a Picture, Mathematical Problem Solving*. The preliminary sources which will be used in the literature search will be ERIC, and the Curriculum Library at the University of Victoria.

Research Design

The research design is quasi-experimental, with pretest/posttest on control and treatment groups. The independent variable is specialized instruction on the use of relational graphs to solve mathematical problems. The dependent variables are: performance on mathematics achievement tests, student problem solving strategies and student attitude towards mathematics and theoretical computer science. Students in four grade six classes will participate in the study, drawn from two middle schools in School District 69. One treatment and one control group will be selected from each school. All groups will be given a pretest consisting of approximately 10 math problems which could be solved using either relational graphs or other strategies. The treatment group will participate in approximately five lessons on how to use relational graphs to solve mathematical problems, and the control group will not receive this treatment. Following the treatment, both control and treatment groups will complete a posttest which, like the pretest, will consist of 10 problems which could be solved using either relational graphs or other problem solving strategies. The questions on the posttest will be modified slightly (variable name, numerical values) so that they are not exactly the same as the pretest, and the order of the questions may vary as well.

mathematical problems, and school demographic. The dependent variables in this study are: mathematics achievement, mathematics and theoretical computer science attitude, and problem solving strategies.

Methods of Data Collection

The primary method of data collection will be pretests and posttests. Participants (both control and treatment group members) will complete both pretests and posttests to measure the use of relational graphs to solve problems. In-class interventions will be recorded, using video recording equipment. These recordings will be stored for later analysis. Participants' comprehension and attitude will be measured using “clickers” at the end of each treatment. Participants will be asked to respond to multiple choice questions using remote “clickers”, handheld devices which allow students' responses to be viewed in realtime on a computer screen. A small number of participants (both from the treatment and control groups) will be selected to participate voluntarily in additional interviews. In these interviews, participants will work with the researcher. The participants will be asked to complete a single problem solving activity, similar to one of the problems presented in the pretest and posttests. The student's progress will be recorded using a video recorder. Once the participant is finished (either having completed the exercise, or out of time), they will be asked to watch the recording of their efforts in solving the problem. The interviewer will ask questions to help the participant clearly communicate: various problem solving strategies they tried; any difficulties they may have encountered; how they coped with difficulties; and any successes they had.

The validity of the pretest and posttest measures will be established by: deriving questions from existing grade six math textbooks; basing the questions on activities designed for middle school students; or ensuring that the questions meet existing Prescribed Learning Outcomes (PLOs) [1,2] and released items from the Foundation Skills Assessment (FSA). All questions will be aligned with the most current grade six curriculum guidelines as defined by the BC Ministry of Education. All problems will be solvable by *both* the use of relational graphs, and by other problem solving strategies. The use of treatment and control groups at both the participating schools will address any validity issues arising from demographic differences between the two school catchments. Reliability will be addressed with inter-rater agreement on the marking of all tests (i.e., where raters review, discuss and reach consensus whenever a discrepancy arises), consistency of the questions on the tests, and test stability. Where more than one rater is used to mark questions on the test, a measure of inter-rater reliability will be clearly measured and given. All questions will be chosen so that they are consistent, that is, they all measure problem solving and the application of problem solving strategies. Prior to the study, the tests will be piloted and refined. The “clicker” data gathered from the treatment group does have weaknesses in terms of validity, as the responses to these questions will be viewable in real-time by all the participants, and while the students' names will not be associated with the responses, total anonymity will not be possible in the classroom. This may affect the results of the questions.

The problem given to the participants of the interview process will align with the problems on the pretest and posttest, and will therefore be a valid test. Reliability will be addressed by the fact that all interviews will be performed by the researcher, so no interviewer training with be required.

In order to address the validity of the analysis of observational data, an external researcher, without prior knowledge of the participants' backgrounds, will perform the analysis of observational data. In addition, the external researcher will be given narrow, specific descriptions of behaviour to be catalogued. The use of video recordings for the gathering of observational data addresses the reliability of this measure. In the case where more than one observer is used to analyse the data, inter-observer agreement will be carefully measured and disclosed.

Data Analysis and Procedures

Participant achievement on the pretest and posttest is a continuous measure. Therefore a t-test will be the primary statistical technique used to analyse this data. The same holds for the measurement of time required for participants to complete the pretest and posttest. The participants' ability to use relational graphs to solve the problems on the pretest and posttest, and participants' responses to the clicker tests are both non-continuous measures, therefore chi-square tests will be used to analyse the data.

Video data gathered during the final few interventions, and the individual tests will be analysed to identify the frequency of specific participant problem solving strategies. The participants' attitude towards problem solving and computer science will be observed. In addition, the video data will be analysed to determine the specific ways in which participants communicate their problem solving strategies.

Ethics and Human Relations

This study poses little if any threat to the participants. It will be made clear to all participants that the testing and observational procedures will have no impact whatsoever on their grade in class. Furthermore, as the pretest and posttest questions will be based either on existing questions from grade six text books, or on problems which target curriculum elements for grade six, the tests will not cause undue stress on the participants. As the researcher has no prior relationship with the participants, there is no “power over” relationship. This study will require approval from both the University of Victoria's Ethics Review Board, and School District 69's review board.

The researcher will gain entry into the proposed research setting (School District 69 middle schools) through her existing relationship with teachers and the district principal of the school district in question. Neither the aforementioned teachers, nor their students, will participate in the study. The district principal has expressed a willingness to participate, and has volunteered to recruit grade six teachers into the study.

References

1. British Columbia Ministry of Education. *Mathematics K to 7: Integrated Resource Package 2007.* Victoria, BC. Queens Printer, 2007.
2. British Columbia Ministry of Education. *Prescribed Learning Outcomes: Curriculum Organizer.* Victoria, BC. Queens Printer, 2007.
3. Denner, J., Werner, L. (2007). *Computer Programming in Middle School: How Pairs Respond to Challenges.* Journal of Educational Computing Research. Volume 37(2).
4. Hart, M., Early, J. P., Brylow, D. (2008). *A Novel Approach to K-12 CS Education: Linking Mathematics and Computer Science.* SIGCSE '08.
5. Higgens, K. M. (1997). *The effect of year-long instruction in mathematical problem solving on middle-school students' attitudes, beliefs, and abilities.* Journal of Experimental Education. Volume 66, Issue 1.
6. Hoyles, C., Kuchemann, D. (2003). *Students' Understandings of Logical Implication.* Educational Studies in Mathematics. 51.
7. Lindh, J., Holgersson, T. (2006). *Does lego training stimulate pupils' ability to solve logical problems?* Computers & Education. Volume 49.
8. Montague, M., Applegate, B. (2000). *Middle School Students' Perceptions, Persistence, and Performance in Mathematical Problem Solving.* Learning Disability Quarterly. Volume 23, No. 3.
9. Niman, J. (1975). *Graph Theory in the Elementary School.* Educational Studies in Mathematics. Volume 6(3).

A.4 Graph Theory Test

Test 1a

Student Name:______________________

Please answer the following questions in the space provided. Show your work.

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Maggie, Stan and Laurie.

a) Draw a picture of these connections:

b) Who has the most friends?

c) Who has the least friends?

d) Are there any groups of friends who aren't connected to a different group of friends?

e) The class desks are organized in a grid. Beth wants to be able to sit beside the window, and also to sit beside all of her friends. Is this possible?

Test 1a

Student Name:______________________

2. A new airline is starting up in Canada. Your family is planning a trip from Victoria to Ottawa. Here is the ad they ran in the local paper:

Air Canadiana's

Introductory Offer!

Book now and save on these one-way flights*:

Toronto to Ottawa	$100.00	Calgary to Victoria	$100.00	Edmonton to Vancouver	$150.00
Toronto to Vancouver	$300.00	Ottawa to Toronto	$100.00	Edmonton to Calgary	$100.00
Toronto to Calgary	$250.00	Ottawa to Calgary	$250.00	Edmonton to Victoria	$200.00
Calgary to Ottawa	$250.00	Vancouver to Toronto	$300.00	Victoria to Edmonton	$200.00
Calgary to Vancouver	$100.00	Vancouver to Calgary	$100.00	Victoria to Calgary	$100.00
Calgary to Toronto	$250.00	Vancouver to Edmonton	$150.00	Victoria to Vancouver	$100.00
Calgary to Edmonton	$100.00	Vancouver to Victoria	$100.00		

*Limited time offer. Some restrictions apply.

a) What route would be best, if you don't want to have to switch planes very often?

b) How much will it cost, if there are 4 people in your family?

Test 1a

Student Name:______________________

3. BC Hydro wants to install new power lines between the following towns in British Columbia. Power lines are made of copper, and copper prices have gone up recently. Each meter of cable costs $1.00. To save money BC Hydro has decided that it's ok if some of the towns are connected through other towns. Power can run in both directions down the wire.

Find the cheapest way to connect these towns: **Vancouver, Coquitlam, Hope, Abbotsford, Kelowna, Penticton, Chilliwack, Merritt, Princeton.**

Route	Distance in km
Vancouver-Coquitlam	26
Hope-Chilliwack	47
Abbotsford-Coquitlam	50
Vancouver-Abbotsford	68
Kelowna-Penticton	68
Chilliwack-Coquitlam	78
Merritt-Princeton	89
Princeton-Penticton	112
Hope-Merritt	121
Merritt-Kelowna	132
Hope-Princeton	133

a) How many kilometers of cable are needed?

b) How much will the power lines cost?

Test 1a

Student Name:______________________

4. Sally and her friends need new graphing calculators for their math class. To save money, the decide to share calculators. They only need to use these new calculators in their Math 10 class, but they need enough calculators so that each girl has one for her class.
Each calculator costs $49.

Student	Math 10 class time	Classroom
Sally	8:50 to 10:00	105
Lindsay	2:00 to 3:10	105
Courtenay	10:30 to 11:40	106
Petra	1:10 to 2:20	106
Molly	9:30 to 10:40	108
Kyla	1:30 to 2:40	109
Bella	1:00 to 2:10	110

a) What is the smallest number of calculators that they could buy, and still have enough?

b) How much money does each friend have to chip in, if they each pay the same amount to buy the calculators?

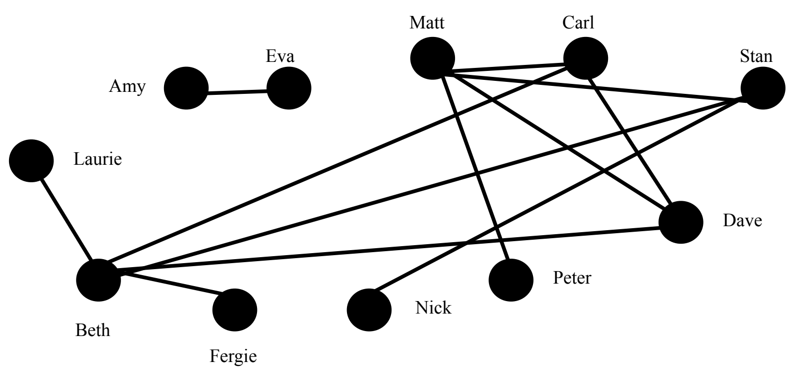
A.5 Graph Theory Test (GTT) Solution

Student ID:____________________

Please answer the following questions in the space provided. Show your work. Spend about 15 minutes on each question. If you get stuck on a question, please explain why you are stuck. You can come back to it if you have time.

1. On the first day of school, Mrs. Polly's math class did a survey of all the students' and their friendships in her new class. She found the following: Amy and Eva are friends. Matt had four friends: Carl, Stan, Dave and Peter. Stan and Nick are also friends, as are Carl and Dave. Beth is friends with Carl, Dave, Fergie, Stan and Laurie.

a) Draw a picture of these connections:



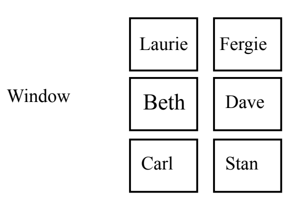
b) Who has the most friends? Beth

c) Who has the least friends? Any of: Amy, Eva, Peter, Laurie, Fergie

d) How many unconnected groups of friends are there in the class? 2 (Amy & Eva, and Group with all other friends)

e) The class desks are organized in a grid. Beth wants to be able to sit beside the window, and also to sit beside all of her friends. Is this possible? Draw a picture to explain your answer.

Yes it is possible



Student ID:_____________________

2. A new airline is starting up in Canada (whose planes run on hydrogen instead of kerosene). Your family is planning a trip from Victoria to Ottawa. Here is the ad the airline ran in the local paper:

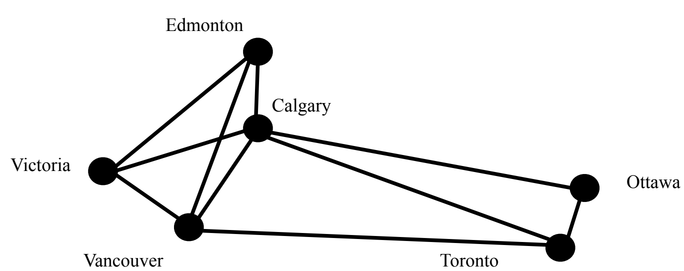
Air Canadiana's

Introductory Offer!

Book now and save on these one-way flights*:

Toronto to Ottawa	$100	Calgary to Victoria	$100	Edmonton to Vancouver	$150
Toronto to Vancouver	$300	Ottawa to Toronto	$100	Edmonton to Calgary	$100
Toronto to Calgary	$250	Ottawa to Calgary	$250	Edmonton to Victoria	$200
Calgary to Ottawa	$250	Vancouver to Toronto	$300	Victoria to Edmonton	$200
Calgary to Vancouver	$100	Vancouver to Calgary	$100	Victoria to Calgary	$100
Calgary to Toronto	$250	Vancouver to Edmonton	$150	Victoria to Vancouver	$100
Calgary to Edmonton	$100	Vancouver to Victoria	$100		

*Limited time office. Some restrictions apply.



a) What route would be best, if you don't want to switch planes very often? List the cities you will have to visit to get from Victoria to Ottawa:

Victoria -> Calgary -> Ottawa

b) How much will it cost (before taxes and fees) to fly one way from Victoria to Ottawa on the route you chose, if there are 4 people in your family?

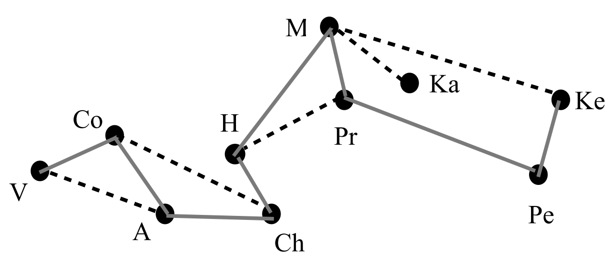
$350.00*4=$1400

Student ID:_____________________

3. TalkTime Corp. wants to install new optical communications lines between the following towns in British Columbia. Each meter of line costs $1.00. To save money TalkTime has decided that it's ok if some of the towns are connected through other towns. Communications can run in both directions down the optical communications lines.

Find the cheapest way to connect these towns together: **Vancouver, Coquitlam, Hope, Abbotsford, Kelowna, Penticton, Chilliwack, Merritt, Princeton.**

Route	Distance in km
Vancouver-Coquitlam	26
Abbotsford-Chilliwack	40
Hope-Chilliwack	47
Abbotsford-Coquitlam	50
Vancouver-Abbotsford	68
Kelowna-Penticton	68
Chilliwack-Coquitlam	78
Merritt-Kamloops	87
Merritt-Princeton	89
Princeton-Penticton	112
Hope-Merritt	121
Merritt-Kelowna	132
Hope-Princeton	133



How to arrive at solution:

- Draw graph of all towns and collections
- One at a time add the cheapest edge so long as it does not create a cycle
- continue until all listed towns are connected
- add distance of all added edges

(Dashed lines are not included)

Edges are added as follows:
V-Co 26
A-Ch 40
H-Ch 47
A-Co 50
~~V-A 68*~~
K-Pe 68
~~Ch-Co 78*~~
~~M-Ka 87**~~
M-Pr 89
Pr-Pe 112
H-M 121
(*not included because they create a cycle)
(** not included because Kamloops not in list)

a) How many kilometers of optical line are needed?

553kms (also accepted $640, even though Kamloops not in list)

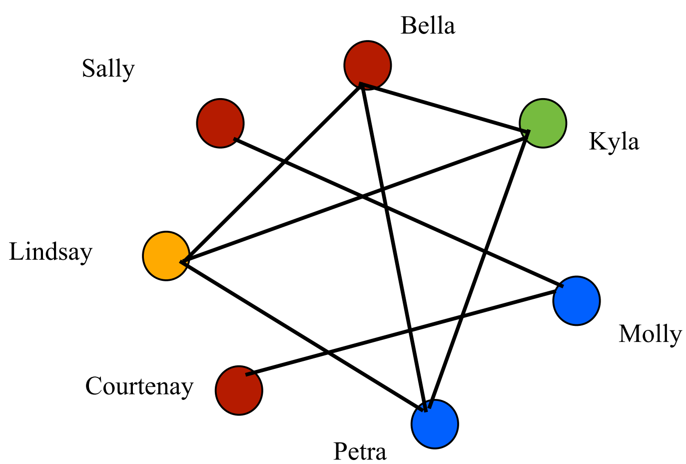
b) How much will the optical lines cost?
553km * $1000/km = $553,000 (also accepted was $640,000)

Student ID:______________________

4. Sally and a group of her friends need new graphing calculators for their mathematics class. To save money, they decide they should buy some calculators to share. They only need to use these new calculators in their Math 11 class, but they need enough calculators so that each girl has one for her class. Each calculator costs $49.

Student	Math 10 class time
Sally	8:50 to 10:00
Lindsay	2:00 to 3:10
Courtenay	10:30 to 11:40
Petra	1:10 to 2:20
Molly	9:30 to 10:40
Kyla	1:30 to 2:40
Bella	1:00 to 2:10

How to solve problem:
- create a vertex for each student
- connect all vertices with an edge if their schedule's conflict (overlap)
- colour the vertices so that no neighboring vertices are the same colour
- each colour represent a calculator needed



a) What is the smallest number of calculators that they could buy, so that every friend has a calculator to use for her mathematics class?

4

b) How much money does each friend have to chip in, if they each pay the same amount to buy the calculators?

(4* $49.00)/7 = $28.00

A.6 GTT Rubric

ID	Question	min	max	scale
1A1	Draw a Picture?	0	2	0 = no, 1 = not a graph, 2 = a graph
1A2	Picture Correct?	0	4	0 = no, ... , ? = correct
1B	correct?	0	4	4 if correct, -1 for delta from correct, min = 0
1C	correct?	0	4	4 if correct, -1 for delta from correct, min = 0
1D	correct?			number given
1E1	possible?	0	1	0=no written answer, 1=wrote an answer
1E2	Draw a picture?	0	1	0 = no, 1 = yes
1E3	picture correct?	0	4	4 if correct, -1 for delta from correct, min = 0
2A1	Draw a Picture?	0	2	0 = no, 1 = not a graph, 2 = a graph
2A2	best route?	0	3	3 = best, -1 for each additional stopover
2B	correct arithmetic	0	1	1=correct, 0 = incorrect
3A1	Draw a Picture?	0	2	0 = no, 1 = not a graph, 2 = a graph
3A2	picture correct?	0	4	4=best, -1 for each error
3A3	km optional	0	2	2= correct, 1 = minor error, 0= incorrect
3b	cost correct	0	2	2= correct, 1 = minor error, 0= incorrect
4A1	draw a Picture?	0	2	0 = no, 1 = not a graph, 2 = a graph
4A2	picture Correct?	0	4	4=best, -1 for each error
4A3	#correct?			number given
4B	cost correct	0	2	2=correct, 1=minor error, 0 =incorrect

A.7 Lesson Plans

A.7.1 Lesson One Part One

Lesson 1: What is a relational Graph?

Objectives:
- students should learn the basic concept of a relational graph
- students should learn to identify basic elements of relational graphs
- students should learn to distinguish between correct and incorrect representation of relational graphs

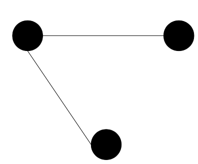
The Lesson:

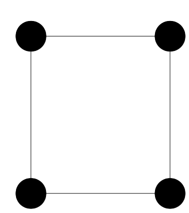
To start, ask students to name some Graphs that they are already familiar with. Some examples include:
- Bar graphs
- Scatter plots
- Line graphs

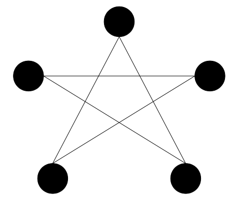
These graphs allow us to represent data using pictures. Graphs can make it easier to see patterns and trends. Graphs can also clarify data by grouping like values together.

Today we're going to learn about another kind of graph, called a Relational Graph. Relational graphs also let us represent data with a picture. But unlike the other graphs, relational graphs can be especially useful for showing Relationships or Connections.

Relational Graphs:







A.7.2 Lesson One Part Two

Lesson 1: Creating Graphs from Ideas

Objectives:

- students will be able to draw a relational graph which represents a set of relationships or connections which are described either verbally or in writing
- students will be able to interpret a relational graph and describe it in terms of relationships or connections either verbally or in writing

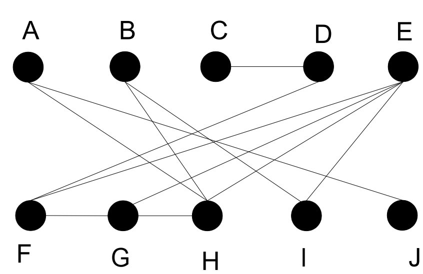
The Lesson:

When trying to understand or describe relationships or connections, it can help to draw a picture. Relational graphs are one kind of picture that can be used in this way.

Example 1:
(Do this example as a class exercise)
There are 10 boys in a classroom. It is the first day of class, and not everyone knows each other yet. The teacher hands out a survey, asking each boy to list the boys (in the class) who are their friends. Draw a relational graph which represents the friendships of these 10 boys.

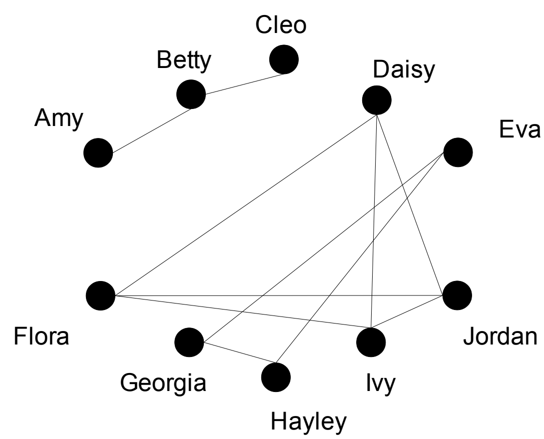
Name	**Friends with:**
Albert	John and Harry
Brian	Harry and Ivan
Cory	Drake
Drake	Cory, Fred
Evan	Harry, Fred, George and Ivan
Fred	George, Evan and Drake
George	Fred, Harry and Evan
Harry	Albert, Brian, Evan and George
Ivan	Brian and Evan
John	Albert

A Solution:



In this solution, each boy is represented by a Vertex (in this case to save room, just their first initial is used to label the vertices. The friendships are represented by edges, connecting the boys who claim to be friends.

Example 2:
(Again, this example can be done as a class exercise)
Given the following graph of friendships, describe in words (using a list or table if you wish) the friendships that it represents:



Solution

Name	**Friends with:**
Amy	Betty
Betty	Amy and Cleo
Cleo	Betty
Daisy	Flora, Ivy and Jordan
Eva	Georgia and Hayley
Flora	Daisy, Ivy and Jordan
Georgia	Eva and Hayley
Hayley	Georgia and Eva
Ivy	Jordan, Flora and Daisy
Jordan	Ivy, Flora and Daisy

A.7.3 Lesson Two

Lesson 2 (Teacher version): Relational Graph Properties

Objectives:

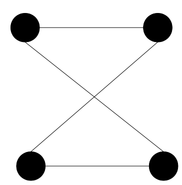
- Students will be able to identify and create isomorphic, planar and plane graphs

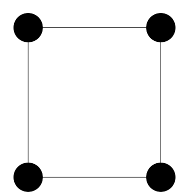
The Lesson:

If they haven't done so yet, have students complete the challenge excercise from Practice Sheet 1b. Once they are done, have them compare their graphs to see if they look the same. If they don't, what does this mean?

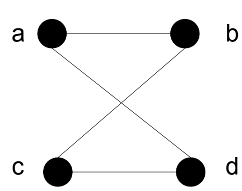
One of the properties of some relational graphs is *isomorphism*. Two graphs are isomorphic if they contain the same relational information. Isomorphic graphs may appear different, but as long as they contain the same information, they are the same.

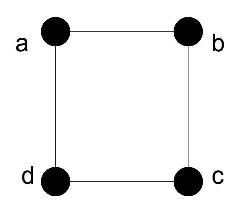
For example, these graphs,





can be labelled, to show they are the same graph.
Is there is some way to untangle the graphs so that you can see that they are infact the same? In this example graph, you could untangle the graph on the left to make it match the graph on the right.





For some graphs, it might not be so easy.
We can see that in both graphs:

- *a* is connected to *b* and *d*
- *b* is connected to *a* and *c*
- *c* is connected to *b* and *d*
- *d* is connected to *a* and *c*

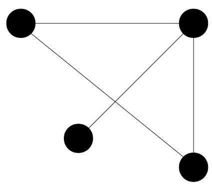
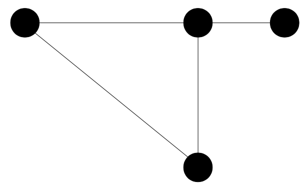
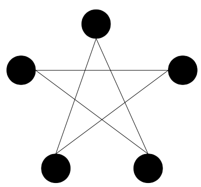
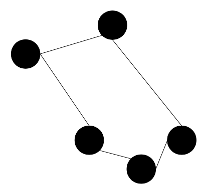
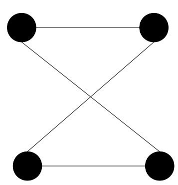
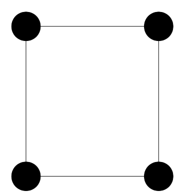
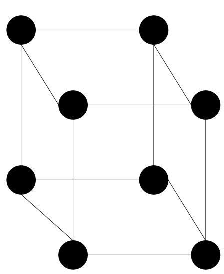
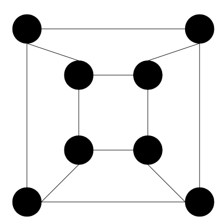
Another way to see this is to list all the Edges. We can name an Edge (if the vertices are labelled) by using the names of the Vertices that it connects to. In the graphs above we have the following Edges: *ab, bc, cd, da*. Note that the edge *ab* is the same as the edge *ba*.

To see if two graphs are isomorphic, try to find a way to label them and end up with the same Edge List. To save time, first ask if:

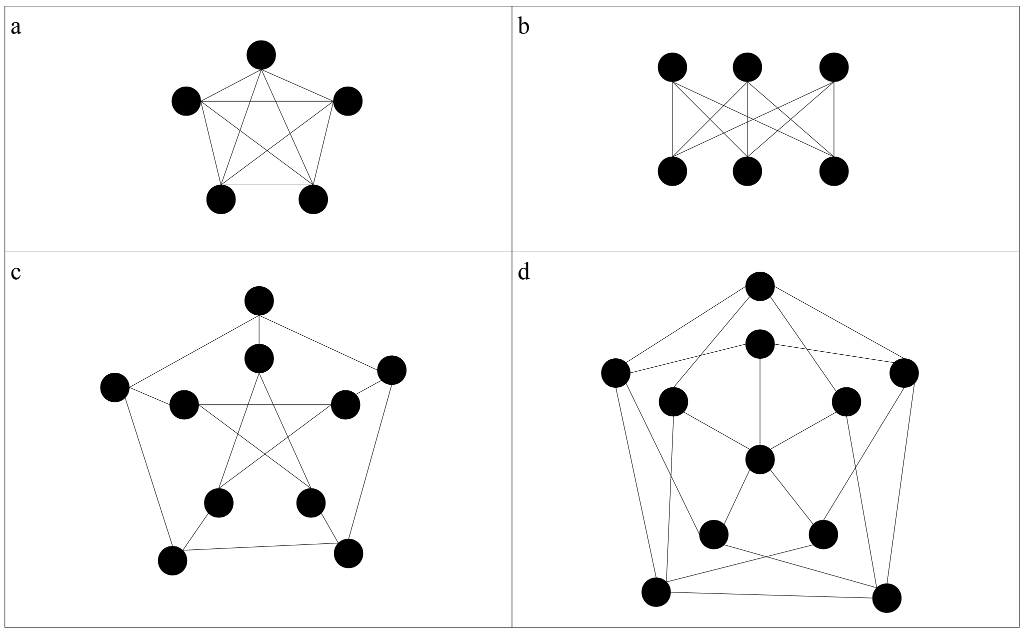
Planarity

Another property of relational graphs is **PLANARITY**. A relational graph is **planar** if you can draw it so that no edges cross. A relational graph which is drawn with no edges crossing is called a **plane graph**.

Here are some planar graphs, along with their plane graphs:

Planar Graph	Plane Graph
	
	
	
	

Not all relaional graphs can be drawn as plane graphs. These relational graphs cannot be drawn as plane graphs:



Do class exercise: Group planarity overhead, and discuss each graph.

Now have the students try Practice Sheet 2b.

A.7.4 Lesson Three

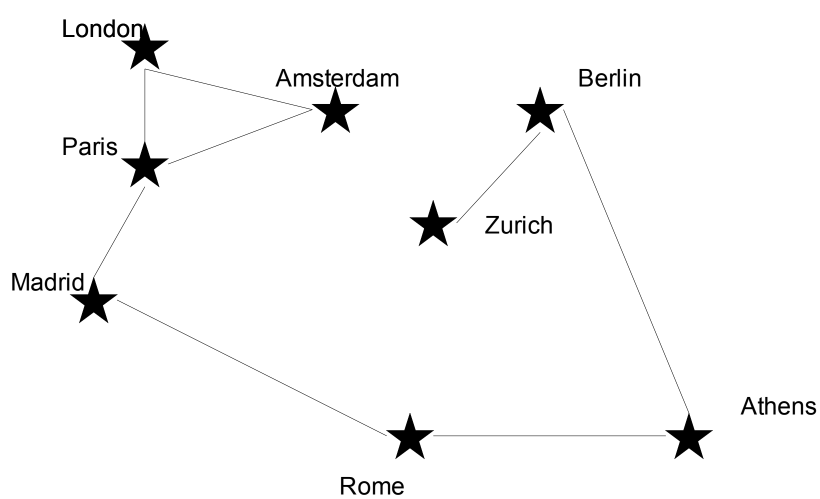
Lesson 3 (Teacher version): Using Relational Graphs to Solve Problems I
Social Networking and MST problems

Objectives:

- Students will be able to use a relational graph which maps onto a physical concept to *solve problems*
- Students will be able to generate a relational graph from a word description

The Lesson:
Relational graphs can help us interpret relationhips between people, places or things.

Minimum Spanning Trees. A Tree is a special kind of relational graph, which has no cycles. A cycle is like a loop.



In this example, there is a cycle from London to Paris to Amsterdam back to London. We could get rid of this loop by removing one of those three edges.

In the next problems we will be creating Minimum Spanning Trees. A minimum spanning tree is the cheapest tree which can connect all the vertices in a graph.

Do clicker activity to identify trees, cycles, and connected graphs.

Step 2: Think about how we will solve the problem:

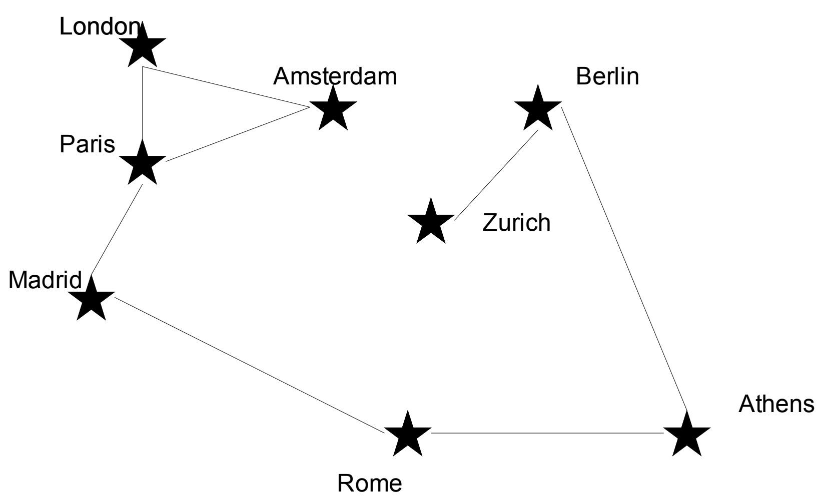
1. How do we make it the cheapest (shortest total length of track)?
2. How do we decide which connections to add?
3. How many connections do we need in total?
4. What is the cost of the system you come up with? Is there a cheaper way?

Note: If time allows, you can let the students try to come up with the cheapest way, in groups of 2 or 3, and see which group gets the best solution. This can help motivate the next stage:

Now we will use an algorithm called a *Greedy Algorithm,* which if done correctly will **always** give the cheapest solution:

1. Pick the cheapest connection which connects two cities, *unless* it creates a cycle. A cycle means you could follow a path on the graph, without going over an edge more than once, and end up where you started.
2. Once each city is connected, STOP

Here is an example of a solution with a cycle:



You can travel from London to Amsterdam to Paris and back to London. This is a *cycle*.

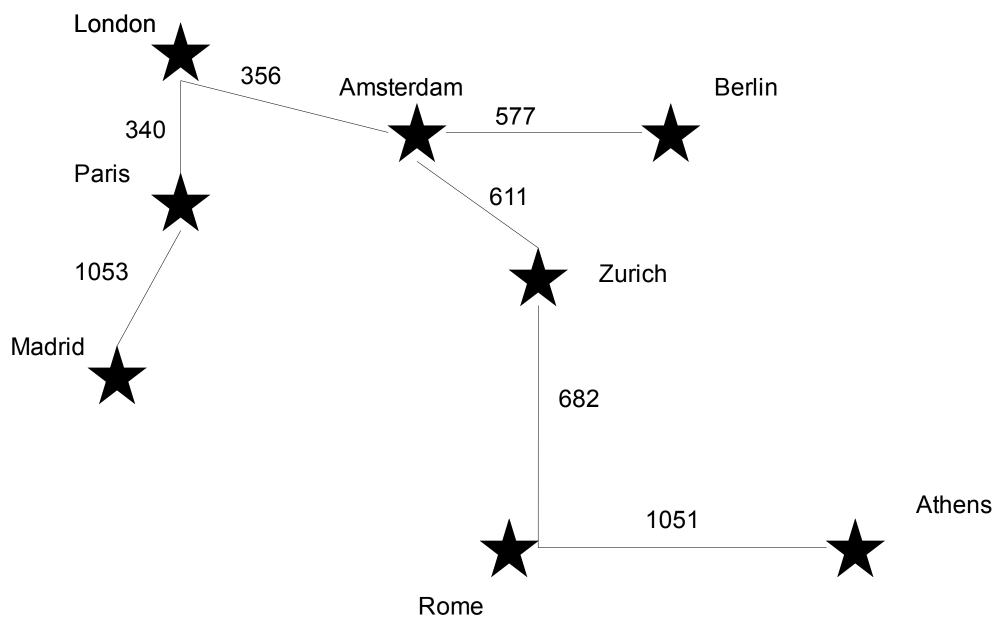
This is called a GREEDY ALGORITHM because at each step we do the *greedy* thing, and add the link that costs the least (unless it adds a cycle).

If we follow this greedy algorithm to solve this problem, here are the links in the order that we consider (an possibly add) them. The ones that are crossed out, are not added because they introduce a cycle. We stop when all the cities are connected.

1. London to Paris: 340
2. London to Amsterdam: 356
3. Berlin to Amsterdam: 577
4. Amsterdam to Zurich: 611
5. ~~Paris to Zurich: 650~~
6. Rome to Zurich: 682
7. ~~Paris to Amsterdam: 748~~
8. ~~London to Zurich: 778~~
9. ~~Paris to Bertlin: 877~~
10. ~~London to Berlin: 935~~
11. Athens to Rome: 1051
12. Madrid to Paris: 1053

Done!

Here is what it looks like:



Total length of track is: 4670km and the total cost is: 4670km x $100/km = $4670000.

NOTE: What we have constructed here is called a *Minimum Spanning Tree.* A Tree is a special kind of relational graph that has no cycles, and where all the nodes are connected. A minimum spanning tree is a tree that is "best". In this case we used *cost,* but we could have used *distance,* or some other measure.

Have students complete Practice Sheet 3.

A.7.5 Lesson Four

Lesson 4 (Teacher version): Using Relational Graphs to Solve Problems II
Shortest Path Problems and Social Networks

Objectives:
- Students will be able to use a relational graph which maps onto a physical concept to *solve problems*
- Students will be able to generate a relational graph from a word description

Social Networks

Draw a graph on the board of the relationships between some fictional friends. Ensure that there a number of distinct connected components, and at least one unconnected vertex.
Discuss the concept of a cluster. A cluster is a connected component which is not connected to another connected component.
Pose the following questions.

Questions and Answers:
1. Who has the most friends? *This is the same as asking which Vertex has the most edges connected to it, or which Vertex has the highest Degree.*
2. Who has the least friends? *This is the same as asking which Vertex has the least edges connected to it, or which Vertex has the lowest Degree.*
3. Find a cluster of friends.
4. How many clusters are there in this class?
5. Can you find a pair of students who aren't friends, who if they became friends would merge two clusters into one? *In general select two clusters and pick one student from each cluster.*
6. How many friendships need to be added so that there is only one cluster in the class?

Shortest Path Problems:

Sometimes it is useful to find the shortest path between two vertices in a relational graph.

A path is a series of: vertex, edge, vertex, edge, ... vertex.
In a shortest path, you will never visit a vertex more than once in a path. It has no cycles.

By shortest, we mean the *cheapest* route. Add up the value of all the edges travelled, to give the cost of the route.

Draw a graph on the board, and have the class try to find the shortest path between two points. Have them list the path (vertex a to vertex b ...)

A.7.6 Lesson Five

Lesson 5 (Teacher version): Using Relational Graphs to Solve Problems III
Colouring and Scheduling

Objectives:
- Students will be able to use a relational graph which maps onto an abstract concept to *solve problems*

Grade Six PLO addressed:

The Lesson:

In this lesson, we'll learn a way to solve conflicts (like you might find when trying to share resources) by transforming a problem into a *different* graph problem

Consider this example:

Your school is planning a field trip to Vancouver. All students will stay at the same hotel, and on wednesday they will go on some sort of tour, then return to the hotel. Each class is planning on visiting a different sight:
- Class A: Museum
- Class B: Granville Island
- Class C: Aquarium
- Class D: Sea Wall
- Class E: Grouse Mountain

All of these places have different hours during which the students can get a tour, and the school wants to save money and only rent as many buses as needed to transport each class to their location.

Venue	Tour hours
Museum	9:00am to 12:00pm
Granville Island	1:00pm to 3:00pm
Aquarium	10:00am to 2:00pm
Sea Wall	10:00am to 11:30am
Grouse Mountain	2:30pm to 4:00pm

Obviously the school could just rent five buses, but is there a cheaper way? Can they use fewer buses?

We assume that if a bus drives a class to a site, then it waits there for them, and takes then back to the hotel.

A.8 Crosstabulation Results

Table A.1: Q4 Draw a Graph

T1a			Posttest 1			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	3	1	0	4
		% Total	37.5	12.5	0.0	50.0
	Picture	Count	1	0	2	3
		% Total	12.5	0.0	25.0	37.5
	Graph	Count	0	1	0	1
		% Total	0.0	12.5	0.0	12.5
	Total	Count	4	2	2	8
		% Total	50.0	25.0	25.0	100.0
T2a			**Posttest 2**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	7	2	1	10
		% Total	63.6	18.2	9.1	90.9
	Picture	Count	0	1	0	1
		% Total	0.0	9.1	0.0	9.1
	Graph	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Total	Count	7	3	1	11
		% Total	63.6	27.3	9.1	100.0
C2a			**Posttest 1**			
			No Picture	Picture	Graph	**Total**
Pretest	No Picture	Count	9	1	0	10
		% Total	81.8	9.1	0.0	90.9
	Picture	Count	0	1	0	1
		% Total	0.0	9.1	0.0	9.1
	Graph	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	Total	Count	9	2	0	11
		% Total	81.8	18.2	0.0	100.0

Table A.2: C1 & C2: Pretest Q1 DAG Correctness

			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	2	3	1	6
		% Total	15.4	23.1	7.7	46.2
	2.0	Count	3	2	2	7
		% Total	23.1	15.4	15.4	53.8
	Total	Count	5	5	3	13
		% Total	38.5	38.5	23.1	100.0
			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	1	0	0	1
		% Total	25.0	0.0	0.0	25.0
	2.0	Count	0	1	2	3
		% Total	0.0	25.5	50.0	75.0
	Total	Count	5	5	3	13
		% Total	38.5	38.5	23.1	100.0

Table A.3: T1 & T2: Pretest Q2 DAG Correctness

			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	1	1	0	2
		% Total	12.5	12.5	0.0	25.0
	2.0	Count	5	0	1	6
		% Total	62.5	0.0	12.5	75.0
Total		Count	6	1	1	8
		% Total	75.0	12.5	12.5	100.0
T2			**DAG**			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	0	3
		% Total	27.3	0.0	0.0	27.3
	1.0	Count	2	0	0	2
		% Total	18.2	0.0	0.0	18.2
	2.0	Count	6	0	0	6
		% Total	54.5	0.0	0.0	54.5
Total		Count	11	0	0	11
		% Total	100.0	0.0	0.0	100.0

Table A.4: C1 & C2: Pretest Q2 DAG Correctness

C1			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	3	0	0	3
		% Total	75.0	0.0	0.0	75.0
	2.0	Count	1	0	0	1
		% Total	25.0	0.0	0.0	25.0
Total		Count	4	0	0	4
		% Total	100.0	0.0	0.0	100.0

C2			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	8	0	0	8
		% Total	61.5	0.0	0.0	61.5
	2.0	Count	5	0	0	5
		% Total	38.5	0.0	0.0	38.5
Total		Count	13	0	0	13
		% Total	100.0	0.0	0.0	100.0

Table A.5: Posttest 1 Q2 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	2	1	2	5
		% Total	11.8	5.9	11.8	29.4
	2.0	Count	7	0	5	12
		% Total	41.2	0.0	29.4	70.6
Total		Count	9	1	7	17
		% Total	52.9	5.9	41.2	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	1	0	5	6
		% Total	12.5	0.0	62.5	75.0
	2.0	Count	1	0	1	2
		% Total	12.5	0.0	12.5	25.0
Total		Count	2	0	6	8
		% Total	25.0	0.0	75.0	100.0

C1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	1	0	0	1
		% Total	20.0	0.0	0.0	20.0
	1.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	2.0	Count	4	0	0	4
		% Total	80.0	0.0	0.0	80.0
Total		Count	5	0	0	5
		% Total	100.0	0.0	0.0	100.0

C2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	0	3
		% Total	13.6	0.0	0.0	13.6
	1.0	Count	5	0	0	5
		% Total	22.7	0.0	0.0	22.7
	2.0	Count	14	0	0	14
		% Total	63.6	0.0	0.0	63.6
Total		Count	22	0	0	22
		% Total	100.0	0.0	0.0	100.0

Table A.6: Posttest 2 Q2 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	1.0	Count	3	0	2	5
		% Total	33.3	0.0	22.2	55.6
	2.0	Count	2	1	1	4
		% Total	22.2	11.1	11.1	44.4
Total		Count	5	1	3	9
		% Total	55.6	11.1	33.3	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	2	0	1	3
		% Total	10.5	0.0	5.3	15.8
	1.0	Count	2	0	5	7
		% Total	10.5	0.0	26.3	36.8
	2.0	Count	4	1	4	9
		% Total	21.1	5.3	21.1	47.4
Total		Count	8	1	10	19
		% Total	42.1	5.3	52.6	100.0

Table A.7: Pretest Q3 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	4	0	1	5
		% Total	50.0	0.0	12.5	67.5
	1.0	Count	0	0	2	2
		% Total	0.0	0.0	25.0	25.0
	2.0	Count	0	0	1	1
		% Total	0.0	0.0	12.5	12.5
Total		Count	4	0	4	8
		% Total	50.0	0.0	50.0	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	6	3	0	9
		% Total	54.5	27.3	0.0	81.8
	1.0	Count	2	0	0	2
		% Total	18.2	0.0	0.0	18.2
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	8	3	0	11
		% Total	72.7	27.3	0.0	100.0

C1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	0	3
		% Total	75.0	0.0	0.0	75.0
	1.0	Count	1	0	0	1
		% Total	25.0	0.0	0.0	25.0
	2.0	Count	0	0	0	
		% Total	0.0	0.0	0.0	0.0
Total		Count	4	0	0	4
		% Total	100.0	0.0	0.0	100.0

C2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	10	0	0	10
		% Total	76.9	0.0	0.0	76.9
	1.0	Count	2	0	1	3
		% Total	15.4	0.0	7.7	23.1
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	12	0	1	13
		% Total	92.3	0.0	7.7	100.0

Table A.8: Posttest 1 Q3 DAG Correctness

T1			**DAG**			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	4	0	10	14
		% Total	23.5	0.0	58.8	82.4
	1.0	Count	0	0	3	3
		% Total	0.0	0.0	17.6	17.6
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	4	0	13	17
		% Total	23.5	0.0	76.5	100.0

T2			**DAG**			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	4	0	2	6
		% Total	50.0	0.0	25.0	75.0
	1.0	Count	0	2	0	2
		% Total	0.0	25.0	0.0	25.0
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	4	0	4	8
		% Total	50.0	0.0	50.0	100.0

C1			**DAG**			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	2	0	2	4
		% Total	40.0	0.0	40.0	80.0
	1.0	Count	0	0	1	1
		% Total	0.0	0.0	20.0	20.0
	2.0	Count	0	0	0	
		% Total	0.0	0.0	0.0	0.0
Total		Count	2	0	3	5
		% Total	40.0	0.0	60.0	100.0

C2			**DAG**			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	21	1	0	22
		% Total	95.5	0.0	4.5	100.0
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	21	0	1	22
		% Total	94.5	0.0	4.5	100.0

Table A.9: Posttest 2 Q3 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	1	0	8	9
		% Total	11.1	0.0	88.9	100.0
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	1	0	8	9
		% Total	11.1	0.0	88.9	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	4	2	10	16
		% Total	22.2	11.1	55.6	88.9
	1.0	Count	1	0	1	2
		% Total	5.6	0.0	5.6	11.1
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
Total		Count	5	2	11	18
		% Total	27.8	11.1	61.1	100.0

Table A.10: Pretest Q4 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	2	2	0	4
		% Total	25.0	25.0	0.0	50.0
	1.0	Count	1	1	1	3
		% Total	12.5	12.5	12.5	37.5
	2.0	Count	1	0	0	1
		% Total	12.5	0.0	0.0	12.5
Total		Count	4	3	1	8
		% Total	50.0	37.5	12.5	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	5	0	0	5
		% Total	45.5	0.0	0.0	45.5
	1.0	Count	4	0	0	4
		% Total	36.4	0.0	0.0	36.4
	2.0	Count	1	1	0	2
		% Total	9.1	9.1	0.0	18.2
Total		Count	10	1	0	11
		% Total	90.9	9.1	0.0	100.0

C1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	0	3
		% Total	75.0	0.0	0.0	75.0
	1.0	Count	1	0	0	1
		% Total	25.0	0.0	0.0	25.0
	2.0	Count	0	0	0	
		% Total	0.0	0.0	0.0	0.0
Total		Count	4	0	0	4
		% Total	100.0	0.0	0.0	100.0

C2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	9	1	0	10
		% Total	69.2	7.7	0.0	76.9
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	2.0	Count	2	1	0	3
		% Total	15.4	7.7	0.0	23.1
Total		Count	11	2	0	13
		% Total	84.6	15.4	0.0	100.0

Table A.11: Posttest 1 Q4 DAG Correctness

T1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	2	3	8
		% Total	17.6	11 8	17.6	47.1
	1.0	Count	5	1	0	6
		% Total	29.4	5.9	0.0	35.3
	2.0	Count	0	1	2	3
		% Total	0.0	5.9	11.8	17.6
Total		Count	8	4	5	17
		% Total	47.1	23.5	29.4	100.0

T2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	1	0	4
		% Total	42.9	14.3	0.0	57.1
	1.0	Count	0	1	0	1
		% Total	0.0	14.3	0.0	14.3
	2.0	Count	0	1	1	2
		% Total	0.0	14.3	14.3	28.6
Total		Count	3	3	1	7
		% Total	42.9	42.9	14.3	100.0

C1			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	1	4
		% Total	60.0	0.0	20.0	80.0
	1.0	Count	0	1	0	1
		% Total	0.0	20.0	0.0	20.0
	2.0	Count	0	0	0	
		% Total	0.0	0.0	0.0	0.0
Total		Count	3	1	1	5
		% Total	60.0	20.0	20.0	100.0

C2			DAG			
			No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	13	2	0	15
		% Total	61.9	9.5	0.0	71.4
	2.0	Count	0	1	0	1
		% Total	0.0	4.8	0.0	4.8
	2.0	Count	3	2	0	5
		% Total	14.3	9.5	0.0	23.8
Total		Count	16	5	0	21
		% Total	76.2	23.8	0.0	100.0

Table A.12: Posttest 2 Q4 DAG Correctness

T1			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	3	0	1	4
		% Total	33.3	0.0	11.1	44.4
	2.0	Count	0	0	1	1
		% Total	0.0	0.0	11.1	11.1
	2.0	Count	0	2	2	4
		% Total	0.0	22.2	22.2	44.4
Total		Count	3	2	4	9
		% Total	33.3	22.2	44.4	100.0

T2			**DAG** No Picture	Picture	Graph	**Total**
Correctness	0.0	Count	8	2	1	11
		% Total	42.1	10.5	5.3	57.9
	2.0	Count	0	0	0	0
		% Total	0.0	0.0	0.0	0.0
	2.0	Count	4	3	1	8
		% Total	21.1	15.8	5.3	42.1
Total		Count	12	5	2	19
		% Total	63.2	26.3	10.5	100.0

Bibliography

[1] Alice. http://www.alice.org/.

[2] Tim Bell, Ian Witten, and Mike Fellows. Computer Science Unplugged. http://www.csunplugged.org/.

[3] Sarah Carruthers. An Interdisciplinary Guide for K-8 Computer Science (CS) Education (poster). In *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education (SIGCSE'10)*, 2010.

[4] Sarah Carruthers. Relational Graphs: What are they? (video). In *Proceedings of the 41st SIGCSE Technical Symposium on Computer Science Education (SIGCSE'10)*, 2010.

[5] Sarah Carruthers, Yvonne Coady, Celina Gibbs, Todd Milford, and Ulrike Stege. *CRYSTAL Pacific Centre for Science, Mathematics and Technology Literacy: Lessons Learned over 5 Years (2005-2010)*, chapter Teaching Problem Solving & Computer Science in the Schools: Concepts and Assessment. SensePublishers, To Be Published.

[6] Sarah Carruthers, Todd Milford, Tim Pelton, and Ulrike Stege. Moving K-7 Computer Science Instruction into the Information Age. In *Proceedings of the Western Canadian Conference for Computing Education (WCCCE)*, 2010.

[7] L. Carter. Why students with an apparent aptitude for computer science don't choose to major on computer science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'06)*, Houston, TX, 2006.

[8] Jan Chipchase. Shared phone use. http://www.slideshare.net/janchip/shared-phone-use.

[9] Douglas H. Clements and Dominic F. Gullo. Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6):1051–1068, 1984.

[10] Pacific crystal. http://www.educ.uvic.ca/pacificcrystal/main.html.

[11] CSTA. A model curriculum for K-12 computer science: Final Report of the ACM task force curriculum committee. Technical report, Computer Science Teachers Associaton, 2003.

[12] A. C. Elliot and W. A. Woodward. *Statistical analysis quick reference guide book with SPSS examples*. Sage, Thousand Oaks, CA, 2007.

[13] ACM/IEEE Joint Task Force. Computing curricula 2005: The overview report. Technical report, Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 2005.

[14] Joyce P. Gall, M. D. Gall, and Walter R. Borg. *Applying Educational Research: A Practical Guide*. Pearson, 2005.

[15] Joanna Goode. Increasing diversity in k-12 computer science: Strategies from the field. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education (SIGCSE'08)*, Portland, OR, 2008.

[16] K. Gunion, T. Milford, and U. Stege. Curing recursion aversion. In *Proceedings of the 14^{th} annual ACM SIGCSE conference on Innovation and Technology in Computer Science Education (ItiCSE)*, Paris, France, 2009.

[17] Katherine Gunion. FUNdamentals of CS: Designing and Evaluating Computer Science Activities for Kids. Master's thesis, University of Victoria, 2009.

[18] Ralph W. Hood, W. Paul Williamson, and Ronald J. Morris. Changing views of serpent handling: A quasi-experimental study. *Journal of Scientific Study of Religion*, 2000.

[19] John Horrigan. Wireless Internet Use. http://www.pewinternet.org-/Reports/2009/12-Wireless-Internet-Use.aspx?r=1, 2009.

[20] C. Hoyles and D. Kuchemann. Students' understandings of logical implication. *Educational Studies in Mathematics*, 51:193–223, 2003.

[21] iClicker. http://www.iclicker.com/dnn/.

[22] Mobile overtakes fixed: Implications for policy and regulation. http://www.itu.int/osg/spu/ni/mobileovertakes/Resources/Mobile-overtakes_Paper.pdf, 2007.

[23] Online statistics. http:www.itu.int/ITU-D/icteye/Indicators/Indicators.aspx, 2007.

[24] S. Keshav. Why cell phones will dominate the internet. *ACM SIGCOMM Computer Communication Review*, 35(2), 2005.

[25] Michael E. Lantz. The use of 'clickers' in the classroom: Teaching innovation or merely an amusing novelty? *Computers in Human Behavior*, 25:556–561, 2010.

[26] LEGO Mindstorms. http://mindstorms.lego.com.

[27] Richard Lehrer, Thomas Guckenberg, and Okhwa Lee. Comparative study of the cognitive consequences of inquiry-based logo instruction. *Educational Psychology*, 80(4):543–553, 1988.

[28] Michael S. Lewis-Beck. *Data Analysis: An Introduction*. Sage, 1995.

[29] J. Lindh and Thomas Holgersson. Does lego training stimulate pupils' ability to solve logical problems? *Computers & Education*, 49:1097–1111, 2007.

[30] Marci Linn. The cognitive consequences of programming instruction in classrooms. *Educational Researcher*, 14(5):14–16+25–29, 1985.

[31] LOGO. http://el.media.mit.edu/Logo-foundation/logo/programming.html.

[32] Ellen Mandinach. Aspects of programming courses that foster problem solving. Technical report, Educational Testing Service, Princeton, NJ, 1986.

[33] John Niman. Graph theory in the elementary school. *Educational Studies in Mathematics*, 6(2), 1975.

[34] British Columbia Ministry of Education. *Mathematics K to 7: Integrated Resource Package 2007*. Queens Printer, Victoria, BC, 2007.

[35] British Columbia Ministry of Education. *Prescribed Learning Outcomes: Curriculum Organizer*. Queens Printer, Victoria, BC, 2007.

[36] The Orchive. http://orchive.cs.uvic.ca/.

[37] Seymour Papert. A computer laboratory for elementary schools. Technical Report 246, National Science Foundation, Washington, D.C., 1971.

[38] Seymour Papert. Interim report of the logo project in the brookline public schools: An assessment and documentation of a children's computer laboratory. Technical report, National Science Foundation, Washington, D.C., 1978.

[39] Seymour Papert and Idit Harel. *Constructionism*, chapter Situating Constructionism. Ablex Publishing Corproation, 1991.

[40] Tim Pelton and Leslee Francis Pelton. Technology outreach workshops: Helping teachers to climb over the technology threshold by engaging their classes. In *Society for Information Technology & Teacher Education International Conference (SITE) 2008)*, Las Vegas, Nevada, 2008.

[41] PicoCricket. www.picocricket.com/.

[42] J. Randolph, G. Julnes, E. Sulinen, and S. Lehman. A methodological review of computer science education research. *Journal of Information Technology Education*, 7:135–162, 2008.

[43] Michael Resnick. Technologies for lifelong kindergarten. *Educational Technology Research and Development*, 46(4), 1998.

[44] Donald F. Roberts and Ulla G. Foehr. Trends in media use. *The Future of Children*, 18(1):11–37, 2008.

[45] Mike Robinson. Robotics-driven activities: Can they improve middle school science learning? *Bulletin of Science Technology Society*, 24(1):73–84, 2005.

[46] Scratch. http://scratch.mit.edu/.

[47] Service Canada Computer Programmers and Interactive Media Developers, 2009.

[48] J. Slonim, S. Scully, and M. McAllister. Crossroads for canadian cs enrollment. *Communications of ACM*, 51(4):66–70, 2008.

[49] SPARCS. SPARCS! The UVic Computer Science Outreach Group.

[50] John Starkweather. Computer science instruction in elementary grades, and exploration of computer-based learning methods. Final report. Technical report, Office of Education, Washington, D.C., 1968.

[51] StarLogo. http://education.mit.edu/starlogo/.

[52] Ulrike Stege. Personal Communication, 2008. During discussion with elementary teachers.

[53] L. W. Stojkiewicz. *Training for Problem-Solving Skills Utilizing a Computer-Assisted Instructional Method*. Phd dissertation, University of California, Berkeley, 1967.

[54] Julie Jepsen Thomas. Reducing anxiety during phase i cardiac rehabilitation. *Journal of Psychosomatic Research*, 39(3):295–304, 1995.

[55] Jeannette Wing. Computational thinking. *Communications of the ACM*, 49(3), 2006.

[56] S. Zweben. Undergraduate Enrollment in Computer Science Trends Higher; Doctoral Production Continues at Peak Levels. Computing Degree and Enrollment Trends from the 2007-2008 CRA Taulbee Survey. Technical report, Computing Research Association, 2008.