

Unit 2:

Problem Solving



Introduction

In order for students to become “computational thinkers” they need experience solving a wide range of problems and the opportunity to experiment with a variety of solution strategies. This unit begins with an introduction to the problem solving process. Students are asked to solve new problems by planning a strategy, designing and producing solutions, and then reflecting on their solutions and strategies.

Throughout the unit the emphasis should be on the process rather than the solution. Most of the world’s problems today do not have single simple solutions. In order to contribute effectively to the solution of these problems, students need to be comfortable in a collaborative environment where multiple approaches are valued and encouraged and where failure is seen as part of the process toward solution. Students must learn to think abstractly and apply known algorithms where appropriate, but also create new algorithms that can be applied to complex problems.

As students reflect on their solution processes and solutions and share those reflections with their peers, it is an opportunity to pull out instances where one strategy might be preferred over another and problems for which there are “standard” solutions versus those where there are many possible solutions.

Many of the problems presented have a mathematical basis and can serve to provide connections between mathematics and computer science. Common computer science topics such as searching, sorting, and graphing are introduced. Although programming the solutions to many of these problems is beyond the scope of this course, students will gain a basic understanding of the algorithms and be able to analyze them. In particular, it is important to emphasize that the models used for solving computational problems are the underpinnings of computer science and as such remain largely the same even as we add new tools and languages.

A key point of emphasis throughout the unit is the connection between the solution “process” and the discussion toward the end of Unit 1 related to how computers are programmed. It is also important to emphasize that not all problems are easily solved by computers.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1-2	Introduce data collection and problem solving.
3	Introduce the four steps of the problem solving process.
4-6	Apply the problem solving process. Use different strategies to plan and carry out the plan to solve several problems.
7-9	Reinforce the four steps of the problems solving process.
10-12	Count in the binary number system. Convert between binary and decimal numbers in the context of topics that are important to computer science.
13-14	Introduce the linear and binary search algorithms.
15-16	Explore sorted and unsorted lists and various sorting algorithms.
17	Introduce minimal spanning trees and how graphs can be used to help solve problems.
18-21	Final projects and presentations

Daily Lesson Plans

Instructional Days: 1-2

Topic Description: This lesson sets the stage for the unit. It provides an overview of data collection and problem solving that will be needed in order to complete the final project.

Objectives:

The student will be able to:

- Recognize various forms of communication as data exchange.
- Describe the implications of data exchange on social interactions.
- Consider privacy of data that they create.
- Explain the difference between data used for making a case and data that informs a discovery.
- Describe good research questions.

Outline of the Lesson:

- Communication Methods and Data Chart (15 minutes)
- Data Journal (15 minutes)
- How data is linked back to us (25 minutes)
- Solving Community Problems Activity (25 minutes)
- Difference between data used for making a case and data that informs discovery (15 minutes)
- Research questions (10 minutes)
- Journal Entry (5 minutes)

Student Activities:

- Discuss Communications Methods and Data Chart with elbow partner.
- Discuss Data Journal with elbow partner.
- Groups complete Solving Community Problems activity.
- Complete journal entry.

Teaching/Learning Strategies:

- Have students share their Communications Methods and Data Chart with their elbow partner.
 - Remind students that this was assigned as homework in Unit 1.
 - Have a few students share “What data is available?”, “Who has access to the data?”
- Data Journal Class Discussion
 - Have students:
 - Compare journals with elbow partners.
 - Write down 5 ways that they give off data.

- Take time to discuss what these data sources might tell us about ourselves if we “aggregated” or collected these data from lots of people – What good might be done? What services might be improved?
 - Think about which of these ways can be linked directly back to them. What are possible implications of the data being linked back to them?
 - Have students read this article about aggregate search data—technically, making search data available to researchers would help improve search engines, but it turns out that search history is intensely personal.
<http://www.nytimes.com/2006/08/09/technology/09aol.html>
 - The Netflix prize is another example. It has recently been cancelled due to FTC concerns over privacy. <http://blog.netflix.com/2010/03/this-is-neil-hunt-chief-product-officer.html>
 - Volunteering data on Facebook and other social networking sites might tell people more about you than you intend.
<http://www.nytimes.com/2010/03/17/technology/17privacy.html>
- Solving Community Problems
 - Present students with a scenario related to the local community. For example, the city council wants to find out about trash disposal in the community in order to clean up the streets.
 - Have students work in groups of 3-4 to outline how they would
 - Approach the problem
 - What kind of data they might need to collect
 - How they would collect and analyze the data
 - Lead a discussion to get at their thoughts. Highlight the differences between making a case and discovery. How would the choice between these determine the kinds of data you would collect?
 - Making a case (advocacy)—Use data to document situations that contribute to make a positive or negative case for something. (e.g., Let the Metro know about timing of trains and buses; tell the principal about something that needs to be done at the school; tell someone about something you’d like to see continued.
 - Discovery—Collect data to document situations and then use the data to learn something. (e.g., could your food choices be improved?; do I always take an efficient route to activities?)
 - What research questions might you ask in each case?
 - What is your research question?
 - Why did you choose to collect these data for this question?
 - What are the limits of this data?
 - What can you confidently say based on your data?
 - What perspectives are left out based on your data?
- Assignment (will be used for the Unit 2 final project)
 - Every day collect data related to where you go after school—location, means of transportation (walk, bike, etc.), how long it takes to get from one location to the next, any other data that you think would be interesting.
- Journal Entry: What are the steps you use to solve a problem?

- Ask students to reflect on whether these steps are the same in all types of problems they solve.

Resources:

- Communication Methods and Data Chart
- Data Journal

Instructional Day: 3**Topic Description:**

This lesson introduces the four main phases of the problem-solving process as defined by G. Polya in [How to Solve It](#).

Objectives:

The students will be able to:

- Name and explain the steps in the problem-solving process.
- Solve a problem by applying the problem-solving process.
- Explain what the word algorithm means.

Outline of the Lesson:

- Candy bar activity (25 minutes)
- Discussion of solutions (10 minutes)
- Introduction of the steps in the problem-solving process (15 minutes)
- Journal Entry (5 minutes)

Student Activities:

- In groups, participate in the candy bar activity.
- Participate in discussion of solutions.
- Reflect on the candy bar activity as it relates to the problem-solving process.
- Complete journal entry.

Teaching/Learning Strategies:

- Candy bar activity
 - Divide the students into groups of 2 or 3. Give each group a candy bar.
 - Explain that their task is to determine how many "breaks" it will take to break the candy bar into 12 equal pieces. One break of one piece of the candy bar will result in that one piece being divided into two pieces. Demonstrate a "break" by breaking the bar into two pieces. Then stack the two pieces together and break or cut the two pieces into four.
 - At this point, have each student write in their journal the number of breaks they think it will take to break the bar into 12 equal pieces. This should be done without talking to their partner or group members.
 - Working together with their partner or group, have the students discuss and then write their plan for solving the problem. They may revise their guess at this point.
 - Once this is completed, the students should implement the plan by opening the candy, breaking the candy, and counting the number of breaks it takes to get 12 equal pieces.
- Discussion of solutions
 - Choose a group to present their plan to the class.

- Sample questions to ask—Was your guess correct? What process did you use to come up with your guess? Did working with your group and creating your plan change your guess? How many breaks did it take (11 is the answer)? Did your plan work?
 - How do the steps they used match what they wrote in their journal?
- Introduction to the steps in the problem-solving process
 - How do the steps they used relate to the “formal” steps of the problem-solving process?
 1. Understand the problem—read or listen to the problem statement.
 2. Make a plan to solve the problem—use pictures, charts, graphs, systematic lists, objects, or act out the solution to help you devise a plan to solve the problem
 - In Computer Science we call this plan an **algorithm**.
 3. Carry out the plan—once the plan is conceived and understood, follow the plan. If you have planned well, this is the easy part.
 4. Review and reflect on how the problem was solved—once the problem is solved, reflect on the plan that was used.
- Extend breaking the candy into N pieces.
 - Post the Number of Pieces/Number of Breaks Chart (without solutions), including N and have students give you the # of breaks needed for each number of pieces.
 - As you go through the chart, ask questions that lead students to the following understandings.
 - One problem-solving strategy used in solving a problem is to solve a problem for specific values, find the pattern and then generalize the solution. In this case, they are generalizing the solution for an unknown positive number of pieces.
- Reflections on the candy bar problem: Ask the students to reflect on the candy bar problem. Why is this problem an important problem to solve for: a carpenter, a chef, a teacher?
- Journal Entry: How is solving this kind of problem the same/different from how you solve a problem in “real life”?
 - Discuss what makes a problem solvable by computer—being able to provide a step-by-step algorithm is one important piece, but context matters. Think back to unit 1 and making a peanut butter and jelly sandwich. Even if we refined our algorithm would a computer be able to make one? No, but a robot could. (Foreshadow Unit 6.)

Resources:

- Polya, G. How to Solve It. 2nd. Princeton, NJ: Princeton University Press, 2004.
Candy bar problem suggested by Dr. Manuel Blum, Carnegie Mellon University
- Candy bars for student groups to use
- Number of Pieces/Number of Breaks Chart

Number of Pieces/Number of Breaks Chart

Number of Pieces	Number of Breaks
1	0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10
12	11
N	<i>N-1</i>

Instructional Days: 4-6**Topic Description:**

Students will apply different strategies to help them make a plan and carry out the plan to solve several problems. These strategies may include (but are not limited to): draw a diagram or picture, make systematic lists, divide and conquer, find the pattern, and guess and check.

Objectives:

The students will be able to:

- Name and explain the steps in the problem-solving process.
- Solve a problem by applying the problem-solving process.
- Express a solution using standard design tools.
- Determine if a given solution successfully solves a stated problem.

Outline of the Lesson:

- Handshake Activity and Fence Post Activity (20 minutes)
- Explanation of solutions (15 minutes)
- Handshake Activity and reflections (75 minutes)
- Presentations of Handshake Activity (40 minutes)
- Discussion of reflections (15 minutes)

Student Activities:

- Work individually on Handshake Activity problem #1 and the Fence Post Activity.
- Volunteers present solutions to problems.
- Work in groups to complete Handshake Activity problem #2.
- Groups give presentations of their problem solutions.
- Discuss reflections on the process.

Teaching/Learning Strategies:

- Handshake Activity problem #1 and Fence Post Activity
 - Students work individually on Handshake Activity problem #1 and the Fence Post Activity.
- Explanation of solutions
 - Have some students volunteer their solutions to the problems.
 - Reinforce each step of the problem-solving process by asking questions similar to those from the candy bar problem. You want students to understand that
 - Diagrams can be very useful in problems like this to look at a smaller version of the problem before trying to solve for N.

- The fencepost problem is a variation of the candy bar problem or the handshake problem.
- Handshake Activity problem #2 and Reflections
 - In groups of 3 or 4, have students discuss, plan, execute, and reflect on Handshake Activity problem #2. Students should follow the directions given in the activity document and write their group's thoughts on paper.
 - Encourage students to make drawings or charts and/or act out the solution. Chart paper can be given to students to display pictures, charts, or graphs. Their job is to explain the process and the solution so that everyone understands.
- Student Presentations
 - Each group should be given about 5-10 minutes (depending on the size of the class) to present their plan and solution to the class. Be sure the students show all 4 steps in the problem-solving process.
 - Students groups should explain their solutions—why they did what they did
- Discussion of reflections
 - Ask students questions that will get them to reflect on why they proceeded in the manner they did. Where did they start? (chart, etc.) What did they do next and why?
 - Is their solution complete enough that it could be given to a computer (if they knew the language the computer was using)? Why or why not?

Resources:

- Polya, G. How to Solve It. 2nd. Princeton, NJ: Princeton University Press, 2004.
- Handshake and Fencepost Activity
- Handshake Activity #2 Sample Solution

Handshake and Fencepost Activity

For each problem, complete the following information.

Understanding the problem:

- What data or information is known?
- What is unknown?
- What are the conditions?

Plan the solution: Show your plan for solving this problem.

Carry out the plan: Using your plan, show your work and your solution.

Review and discuss your solution: Reflect on your solution.

Complete problems #1 and #2 individually.

1. Handshake Problem #1: Assume there are 20 people in a room, including you. You must shake hands with everyone else in the room. How many hands will you shake? If there are N (where $N > 0$) people in the room, how many hands will you shake?
2. Fence Post Problem: You need to build one side of a fence that is 12 yards long. This fence will be built with fence posts and rails that connect one fence post to another. If each fence post is 1 yard away from the next fence post, how many fence posts will be needed for this side of the fence? How many fence posts will be needed for a side of a fence that is N (where $N > 0$) yards long?

Read and begin planning your solution for problems #3 and #4. These problems will be completed in class tomorrow with your group. Each group will present their solutions to the class.

3. Handshake Problem #2: Assume there are 10 people in a room, including you. Each person in the room must shake hands one time, and only time, with all the other people in the room. How many handshakes will occur? If there are 20 people in the room, how many handshakes will occur? If there are N (where $N > 0$) people in the room, how many handshakes will occur?
4. Reflections: Why are problems like these important to learn how to solve? How could this type of solution be of benefit to a carpenter, a chef, a teacher?

Handshake Activity #2 Sample Solution

The sample solution is only one possibility. Student groups may have a wide variety of strategies. Ask questions that probe their understanding of the steps of the problem-solving process they used.

Understanding the problem:

- What data or information is known? *There are 10 people or N people in the room.*
- What is unknown? *Total number of handshakes*
- What are the conditions? *Each person must shake hands only one time with all others in the room. All of the handshakes must be added together.*

Plan the solution: *A sample plan could be to describe the plan in words or use a chart or draw a picture and then act it out.*

Have the people line up in the room. The first person in the line walks down the line and shakes hands with all of the people in the line and then leaves the room. Count the number of handshakes and add to the total.

The next person in line walks down the line and shakes hands with all of the people left in the line and then leaves the room. Count the number of handshakes and add to the total.

This continues until there are only 2 people left. They shake hands and leave together. Increase the total by one.

Once the answer is known for 10 people, look for a pattern. Try the process for 5 people, 2 people. See if the pattern holds.

Carry out the plan: Using your plan, show your work and your solution.

Person	Shakes Hands with # of people left in line
A	9
B	8
C	7
D	6
E	5
F	4
G	3
H	2
I	1
J	0 (last person in line— no one left to shake hands with)

Now add up the number of handshakes: $9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0 = 45$

For 10 people, the answer is the sum of the numbers from 1 to 9, which is 45. 9 is $10 - 1$.

For 5 people, the answer is the sum of the numbers from 1 to 4, which is 10. 4 is $5 - 1$.

For 2 people, the answer is the sum of the numbers from 1 to 1, which is 1. 1 is $2 - 1$.

For N people, the answer is the sum of the numbers from 1 to $(N-1)$.

Review and discuss your solution: Each person shakes hands with $N - 1$ other people. The answer is not $N(N-1)$, though, because each handshake counts as the one handshake for each person, but only one handshake for the total. The Hershey Bar problem helped to start the plan for this problem, but I needed to adjust the plan to only allow one handshake between each pair in the room.

So the 10 people make 9 handshakes each, but each handshake happens between 2 people, and can only be counted once. I could "divide" the handshake and let each person count the handshake as a $1/2$ handshake. So 10 people make 9 half-handshakes each = 45 handshakes.

N people make $N-1$ half-handshakes each = $N(N-1)/2$

The sum of the numbers from 1 to $N-1 = N(N-1)/2$

Instructional Days: 7-9**Topic Description:**

This lesson reinforces the four main phases in the problem-solving process.

Objectives:

The students will be able to:

- Solve a problem by applying the problem-solving process.
- Express a solution using standard design tools.
- Determine if a given solution successfully solves a stated problem.

Outline of the Lesson:

- Cultural background of cornrow braiding (15 minutes)
- Group discussion on cultural background of cornrow braiding (15 minutes)
- Cornrow curves design tool tutorial (80 minutes)
- Cornrow curves project (50 minutes)
- Gallery walk (5 minutes)

Student Activities:

- Work individually to review the history of cornrow braiding.
- Work in groups to answer reflection question and share with the remainder of the class.
- Work with elbow partner to complete the tutorial.
- Work individually to complete cornrow curves project.
- Participate in gallery walk.

Teaching/Learning Strategies:

- Cultural background of cornrow braiding
 - Students read the cultural background and how to braid sections (csdt.rpi.edu, Cornrow Curves).
- Group discussion on cultural background of cornrow braiding
 - Divide students into groups of 3-4 and ask each group to reflect on one of the following sections:
 - African Origins
 - Middle Passage
 - Civil War to Civil Rights
 - Hip Hop
 - Each group shares their response with the rest of the class.
- Cornrow curves design tool tutorial
 - Individual students complete Part I of the tutorial following all instructions and checking their work with their elbow partner.
 - Discuss any issues as a class before proceeding to Part II.
 - Complete Part II of the design tutorial.

- Stress mathematics and structured inquiry.
 - Reinforce concepts such as iteration, dilation, translation.
- Cornrow curves project
 - Each group of students should complete the following:
 - Students create their own design.
 - Describe each step of the problem-solving process used.
 - Highlight the mathematical concepts used and where and how they are used.
 - Reinforce the strategy of finding a similar problem that has already been solved to help solve the new problem.
- Gallery walk of designs
 - Students share their solutions.

Resources:

- Culturally Situated Design Tools Cornrow Curves—csdt.rpi.edu (courtesy Ron Eglash)

Instructional Days: 10-12

Topic Description: This lesson introduces the binary number system and how to count in binary. Students will learn how to convert between binary and decimal numbers in the context of topics that are important to computer science.

Objectives:

The students will be able to:

- Count forward and backward in binary.
- Explain why binary numbers are important in computer science.
- Use binary digits to encode and decode messages.

Outline of the Lesson:

- Journal Entry (5 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (counting in binary) (50 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (binary number system) (50 minutes)
- Revisit journal entry (5 minutes)
- Journal Entry (5 minutes)
- Discussion of why binary numbers are important in computer science (15 minutes)
- CS Unplugged Activity 1: Count the Dots—Binary Numbers (Email and Modems, Counting Higher than 31) (35 minutes)

Student Activities:

- Complete journal entry.
- Participate in the Count the Dots activities.
- Revisit journal entry.
- Complete journal entry.
- Participate in a discussion of why binary numbers are important in computer science.
- Complete Count the Dots activities.

Teaching/Learning Strategies:

- Journal Entry: How high can you count with your ten fingers?
- Use the CS Unplugged: Count the Dots activity to introduce binary representation and counting in binary.
 - Start with the introductory activity on p. 4 of the activity. (The activity can be downloaded from <http://csunplugged.com>) It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to represent numbers and count in binary. What follows is the minimal suggestion.

- Have 5 students come to the front of the room and demonstrate as you follow the instructions and ask the questions. (Each student should receive a large card with one of the numbers of dots—1, 2, 4, 8, 16)
- Use the CS Unplugged: Count the Dots activity to explain the binary number system and have the students practice counting forward and backward.
 - Complete the Binary Numbers activity on p. 5 and Working with Binary activity on p 7.
 - Have 5 students come to the front of the room and try counting as you call out the numbers. (Each student should receive a large card with one of the numbers of dots—1, 2, 4, 8, 16)
 - Have different groups of 5 students at a time come to the front and have the other students provide counting and representation challenges. You could also have a competition with multiple teams of students each trying to get the answer. There are many other possibilities. Be creative!!
- Revisit Journal Entry.
- Journal Entry: Complete the Sending Secret Messages activity on p. 8 of the CS Unplugged: Count the Dots activity. (Solution is on p. 13.)
- Discussion of why binary numbers are important in computer science
- Complete the remaining activities in CS Unplugged: Count the Dots. (Email and Modems—p. 9, Counting Higher than 31—p. 10, and/or More on Binary Numbers—p. 11)

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged. Canterbury, New Zealand: 2002.
- Computer Science Unplugged Activity 1: Count the Dots—Binary Numbers, pp. 3-13
- Binary number cards for each student
- Large binary number cards for the demonstrations

Instructional Days: 13-14

Topic Description: This lesson introduces the linear and binary search algorithms.

Objectives:

The students will be able to:

- Describe the linear search algorithm.
- Describe the binary search algorithm.
- Explain conditions under which each search might be appropriate.

Outline of the Lesson:

- Tower Building Activity (55 minutes)
- Model tower building algorithm. (25 minutes)
- Model binary search (15 minutes)
- Comparison of linear and binary search (15 minutes)

Student Activities:

- In pairs complete the Tower Building Activity.
- Model the tower building algorithm.
- Students participate in the activity modeling binary search.

Teaching/Learning Strategies:

- Tower Building Activity
 - Have students complete the Tower Building Activity with their elbow partner and write their solutions in their journals.
- Model tower building activity.
 - Have students share their solutions with another elbow partner pair.
 - Have one set of students use 10 legos (or checkers or some other easily manipulated piece) to model the algorithm for solving the problem in front of the entire class.
 - Note: The solution is to start by taking half of the height of the tower and create that number of stacks of 2. Continue halving the number of stacks and doubling the height (plus one stack of any remainder) until the desired height is reached. This foreshadows binary search. (See sample solutions.)
- Model binary search.
 - Use 2 copies of the same dictionary. Hand one dictionary to 2 students and have them pick out a word in the dictionary.
 - Choose 2 other students to count the number of times you choose a word from the dictionary to search for the students' word.
 - Start by using a linear search. It should not take long for students to suggest that this is not a good strategy. Ask them to provide a better strategy.

- Guide them to binary search.
 - Discuss the number of guesses required and how this is similar to the tower building problem.
- Comparison of linear and binary search.
 - Linear—start at the beginning, look at each item until you find it or there is no more data. Data can be sorted or not.
 - Binary—look at middle item, eliminate the half where the value is not located. Find the new middle element and continue the process until you find it, or there is no more data. Ask students to describe what is necessary in order to use a binary search—the list must be sorted.
 - Have students provide examples of where each type of search is appropriate and why.
 - Note that decisions often need to be made about whether to maintain lists in sorted order, provide an option for sorting should it be necessary, etc. based on the types of searches that are expected to be performed on the data.

Resources:

- Shasha, Dennis. [The Puzzling Adventures of Doctor Ecco](#). Mineola, New York: Dover Publications, Inc., 1998.
- Tower Building Activity
- Sample Solutions for Tower Building Activity

Tower Building Activity

Donald Trump wants to build a 100 meter high tower as quickly as possible. He has unlimited resources and an unlimited budget and is willing to spend any amount to get the job done.

He has chosen to build the tower with blocks that are 100 meters long and 100 meters wide, but only 1 meter tall. The blocks interlock on top and bottom (like legos). They cannot be stacked sideways.

Using special lifters, putting one stack on top of another stack takes one week regardless of how high the stacks are.

What is the shortest amount of time that it will take to build the tower?

Suggestions:

- Use something like legos or a graph to help solve this problem.
- Start with a smaller tower of 5 or 10—solve a smaller problem.
- Extend that knowledge to the larger problem.

Sample Solutions for Tower Building Problem

5 meter tower

Week #	# of stacks	# of blocks in stack	Remainder
1	2	2	1 stack of 1
2	1	4	1 stack of 1
3	1	5	

Three weeks needed for completion.

10 meter tower

Week #	# of stacks	# of blocks in stack	Remainder
1	5	2	
2	2	4	1 stack of 2
3	1	8	1 stack of 2
4	1	10	

Four weeks needed for completion.

100 meter tower

Week #	# of stacks	# of blocks in stack	Remainder
1	500	2	
2	250	4	
3	125	8	
4	62	16	1 stack of 8
5	31	32	1 stack of 8
6	15	64	1 stack of 40
7	7	128	1 stack of 104
8	3	256	1 stack of 232
9	1	512	1 stack of 448
10	1	1000	

Ten weeks needed for completion.

In general: The number of weeks is the smallest n such that the height of the tower is less than 2^n .

Instructional Days: 15-16

Topic Description: In this lesson the concept of a list (sorted and unsorted) and sorting algorithms will be explored.

Objectives:

The students will be able to:

- Define sorted and unsorted lists.
- Describe various sorting algorithms.
- Compare various sorting algorithms.

Outline of the Lesson:

- Journal Entry (15 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (explore sorting) (30 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (discover and describe sorting algorithms) (30 minutes)
- CS Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms (compare sorting algorithms) (35 minutes)

Student Activities:

- Complete journal entry.
- Groups participate in the various parts of the CS Unplugged: Lightest and Heaviest activity.

Teaching/Learning Strategies:

- Journal Entry: List examples of where it matters whether items are in order (sorted).
 - Have volunteers provide examples from their lists and explain why it matters that they are sorted; in other words, what are the consequences if the list is not sorted?
 - Ask students about the data that they have been collecting. How easy would it be for them to sort their data by hand? Does it get harder to do this with more data? Point out that this is one of the major advantages of computing—the ability to manage large sets of data that could not easily be managed by humans.
- CS Unplugged: Lightest and Heaviest activity
 - The activity can be downloaded from <http://csunplugged.com>. It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to describe the various types of sorting. What follows is the minimal suggestion.
 - Divide students into groups of 3-4 and give each group a set of weights and a balance scale as described in steps 1 and 2 on p. 66 of the Sorting Weights Activity. (There are many possible ways to make the weights. One would be to use bags with varying numbers of pieces of candy. If you don't have balance scales, you can help students come up with a strategy that will

- simulate a scale. For example, if you make the weights clearly different in weight, they could do this by feel.)
- Have students complete #3 and #4 on p. 66 and then discuss their answers as indicated.
 - Have students complete #5 on p. 66.
 - At this point in the activity, students should present their findings to the class and discuss. Point out the selection sort information on p.66.
 - Have students complete the Divide and Conquer activity on p.67. Throughout, guide students as necessary and have them keep track of the processes they use.
- If time permits, have students try both sorting methods to sort cards that have 50 random numbers on them and analyze the number of comparisons required for each.

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged., New Zealand: 2002.
- Computer Science Unplugged Activity 7: Lightest and Heaviest—Sorting Algorithms, pp. 64-70
- Containers of the same size with different weights
- Balance scales

Instructional Day: 17

Topic Description: Minimal spanning trees and graphs will be explored. Students will learn how graphs can be used to help solve problems.

Objectives:

The students will be able to:

- Solve a minimal spanning tree.
- Draw a graph to solve a problem.

Outline of the Lesson:

- CS Unplugged Activity 9: The Muddy City—Minimal Spanning Trees (20 minutes)
- CS Unplugged Activity 9: The Muddy City—Minimal Spanning Trees (extension) (35 minutes)

Student Activities:

- Participate in the various parts of the CS Unplugged: The Muddy City activity.
- Participate in the various parts of the CS Unplugged: The Muddy City activity extension.

Teaching/Learning Strategies:

- CS Unplugged: The Muddy City activity
 - The activity can be downloaded from <http://csunplugged.com>. It will be helpful to read through the entire activity in advance, so that you can revise questions, add your own questions, and think about how you might want to structure each part of the activity. The goal is for students to be actively involved in some way and for all students to be able to describe shortest path strategies. What follows is the minimal suggestion.
 - Follow the directions in The Muddy City Problem on p. 78.
 - Have students work with their elbow partners.
 - Have students share their solutions and lead the follow-up discussion p.77.
- CS Unplugged: The Muddy City activity extension
 - Have students repeat the Muddy City Problem with the abstract representation in the figure on p. 79 and answer the questions on p 79.
 - Discuss various applications of this problem in anticipation of the final project (p.80).
 - Emphasize the idea of shortest path.

Resources:

- Bell, Tim, Ian Witten and Mike Fellows. Computer Science Unplugged. Canterbury, New Zealand: 2002.
- Computer Science Unplugged Activity 9: The Muddy City—Minimal Spanning Trees, pp. 76-80

Instructional Days: 18-21

Topic Description: Students work on final unit project.

Objectives:

The students will be able to:

- Incorporate all unit objectives into the final project.

Outline of the Lesson:

- Explanation of final project (15 minutes)
- Completion of final projects (150 minutes)
- Presentations of final projects (55 minutes)

Student Activities:

- Groups work on final projects.
- Groups present final projects.

Teaching/Learning Strategies:

- Complete Muddy City activity if necessary.
- Explanation of final project
 - Distribute final project explanation.
 - Note: You may wish to modify the scenario of the problem to address student interests and abilities. Another possible example might be a variation on finding the cheapest route between locations based on the price of gasoline. This could be in the context of a family vacation, carpool routes, running errands, etc.
 - Divide students into groups of 3-4.
- Completion of final projects
 - Monitor student work, answering questions as necessary.
- Presentations of final projects
 - Have each group present the information in their final project.
 - Groups respond to questions from other students and teacher.

Resources:

- Mapping website such as www.maps.google.com
- Final Project (This project is adapted from MathmaniaCS Lesson 13 (<http://www.mathmaniacs.org/lessons>)
- Final Project Sample Rubric

Final Project

For this project you will use the data you have been collecting about your activities after school. Each group member should determine the day on which they visited the most locations after school and what those locations were.

Scenario:

Assume that for one day you need to carpool with the other members of the group in order to get to all of the locations you each identified on the day you visited the most different locations. Determine the shortest route in terms of miles and then determine the shortest route in terms of time. Are these the same? Why or why not? What other conditions might you want to consider? (Your data may give you some ideas.)

You will need to use a map in order to calculate the distances. The data you collected has the times. Your presentation can be given as a poster, a PowerPoint, a video or other pre-approved product.

Your presentation should include:

1. The names of people in your group
2. A picture (graph) representing all the locations with all roads between them labeled with mileages and times
3. A detailed plan of your solution
4. A written explanation of the strategies you used to find the shortest route
 - The solution on the graph and the total number of miles
5. A written explanation of the strategies you used to find the least amount of time
 - The solution on the graph and the total time
6. Why the shortest routes are the same or why not
7. Other conditions you might want to consider and what data gave you the idea

Final Project Sample Rubric

Group Members Names: (up to 4)

Do you have?	Points Possible	Yes	No	Points Earned
Detailed Plans				
Overall plan to solve the problem (explanation of strategies)	10			
Data Journal for each group member	20			
Other parts of your project:				
Graph labeled with locations and mileage	10			
Graph labeled with times	5			
Solution labeled on graph	15			
Total mileage and total time	10			
Other conditions to consider	10			
Presentation				
All group members participate	5			
Present all parts of the project	10			
Answer questions from audience	5			
TOTAL:	100			