Teacher's Guide

The Grand-Challenge- Secure Cyberspace:

In the 21st century, networks of electronic information flow play a role in almost all aspects of life. Known as cyberspace, this web of information offers a wide variety of benefits to civilization that have never before existed, most significantly in communication as well as financial systems. Unfortunately, even the most complex and well protected of these same systems can be breached, posing a serious threat to both personal privacy and national security. The goal of Secure Cyberspace Grand-Challenge is to push for a solution to this enormous threat.

Learning Objectives:

The objective of this presentation is to spark 4th and 5th grade student's interest in programming in order have them one day become engineers capable of solving the Secure Cyberspace Grand-Challenge. This will be accomplished by using hands-on activities to demonstrate the importance and usefulness of algorithms in both programming and everyday life. They will be familiarized with the specificity required in many programming languages, which implement algorithms, and the concept of efficient coding.

Technical Background:

This presentation will provide an introductory glance at the development of algorithms and their importance within global society. Algorithms are the methods one uses to obtain a goal. These include anything from the recipes used to create your favorite food dishes, to how your GPS figures out how to get to a certain destination, to how you tie your shoe. Algorithms are directly involved with the grand challenge of securing cyberspace and consist of everything from the cryptographic standards used today, which contain algorithms to scramble messages preventing them from being read, to the methods by which input is securely given to programs, such as how you enter your username and password into websites.. Banks, utilities, corporations and websites are all secured with current encryption algorithms and would be at risk without more advanced security conventions. Current encryption algorithms are constantly being beaten by today's criminals, as attacks become more sophisticated and intelligent. If these algorithms and protocols were to be cracked, the economic toll would be extreme and widespread, giving unlimited access to bank accounts, corporate information, and public utilities. Identity theft and denial of service attacks are already major issues that afflict today's computer systems and cause issues to thousands of Americans every year. To solve these issues, an evolution of our security algorithms is required, increasing their complexity and therefore, the difficulty required to get access to private information. This would involve both individual encryption schemes to protect data flow between networks, as well as more secure firewalls to prevent inner access. With the hard work of future security professionals creating newer, more secure algorithms, these goals can be obtained.

Budget:

Total: $0
(The only costs incurred would be from printing off materials for each student.)

Initial Setup:

1. Separate the kids into groups of two or let them sort themselves if you wish. Sit them down in these pairs.

2. Give each student a copy of the handout in appendices A as well as a writing utensil.

3. Give a basic introduction to algorithms. Ask the kids if they have heard of algorithms before, briefly go over the definition of an algorithm, and see if the kids can come up with some examples of algorithms.

Terms to Know:

bubble sort: *a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order.*

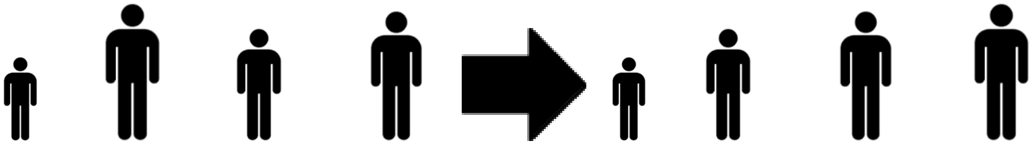
merge sort: *a recursive algorithm that continually splits a list in half. If the list is empty or has one item, it is sorted by definition (the base case). If the list has more than one item, we split the list and recursively invoke a merge sort on both halves.*

Activity 1: Sorting Studnets by Height

Goal: Sort the kids in one line, shortest to tallest (bubble sort)

1. For this initial activity, gather the students in a random single file line, shoulder-to-shoulder. Be sure to make the line as straight as possible in order to avoid confusion.

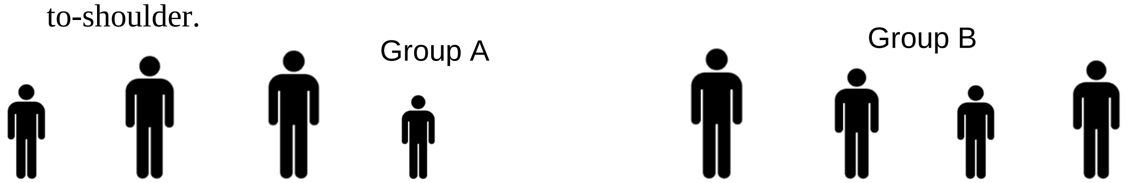
2. Instruct the students to sort themselves in a line that is arranged from shortest to tallest. Each student may only switch places with his/her neighboring classmate. This process can be repeated with new neighbors, and should result in the slow rearrangement of the entire line to the order desired.



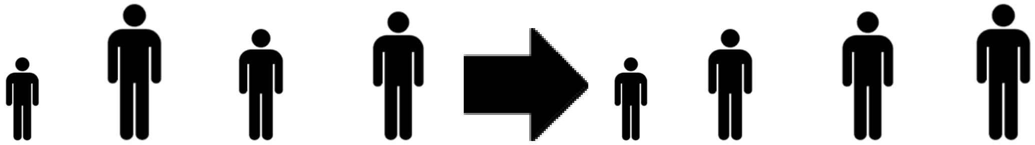
3. Bring to the class's attention how long it took to form the line. Ask them what made the task difficult. Addressing time will prove essential for the second phase of Activity 1.

Goal: Sort the kids in one line, shortest to tallest (merge sort)

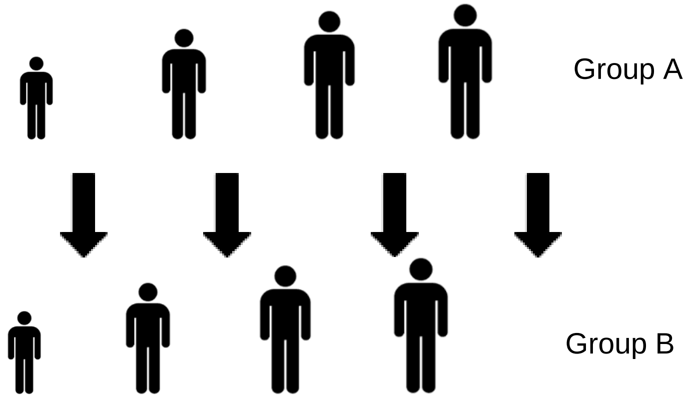
4. Split the class randomly into two groups. A mix of both taller and shorter kids is recommended for each group. Both should form a randomized single file line, shoulder-to-shoulder.



5. Instruct each group to sort themselves in a line that is arranged from shortest to tallest. Each student may only switch places with his/her neighboring classmate. The time taken for the students to organize should be less than that for the single larger line.



6. Have the two organized groups merge. The two shortest students should merge first in such a way that they are next to each other. In this zipper like fashion, proceed to form one line. Final changes in height can be made by switching positions in the same manner described earlier.



7. The "merge sort" method should take less time than the "bubble sword" method. Analyze the different style of organization used with the students and its significance.

Activity 2: Finding the Most Efficient Path

Goal: Let kids find the most efficient path on a map, and emphasize that the most efficient path does not always correlate to the shortest. Rules are stated in Appendices A.

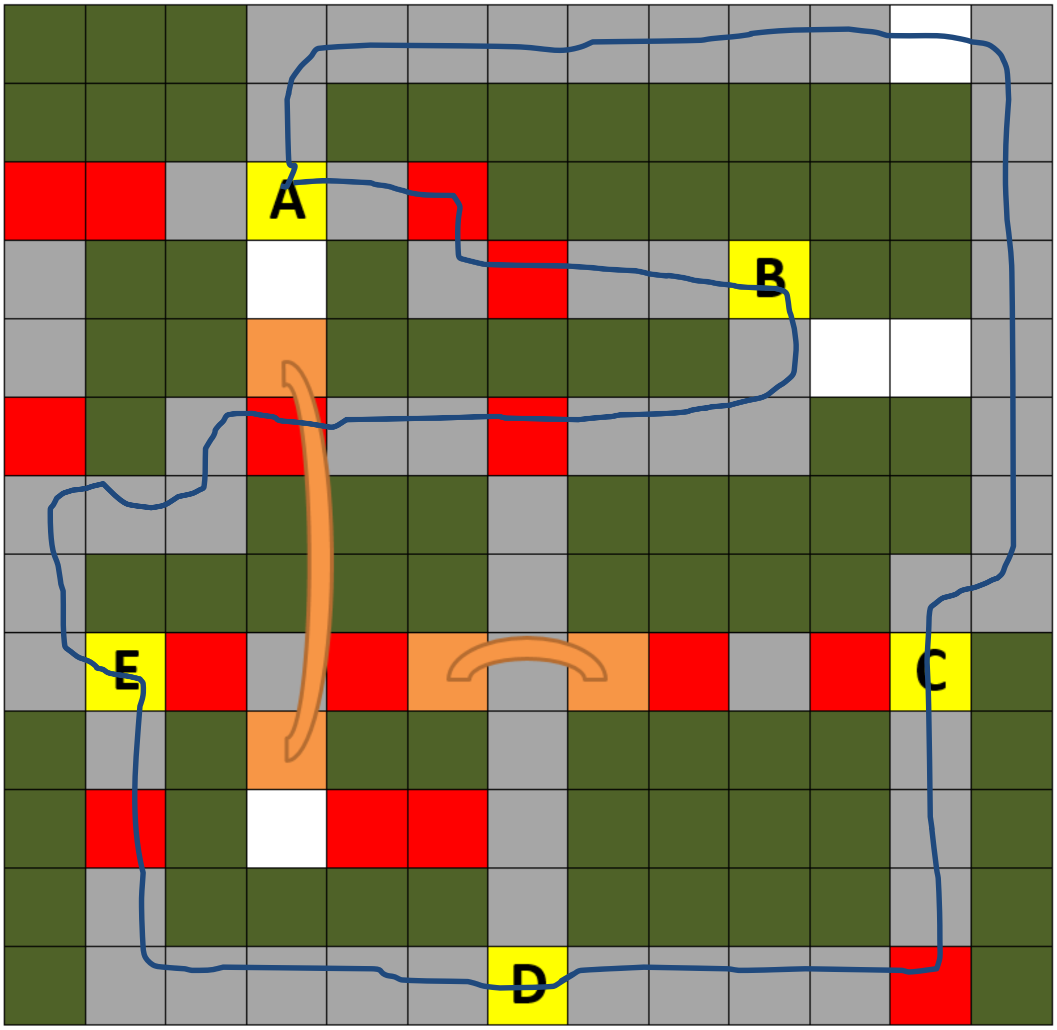
1. Explain to students how different color squares have different fuel efficiency. State that the goal of this activity is to find the most fuel efficient path. Walk through a couple of paths with kids to ensure they understand the task at hand.

2. Let the students search. Be sure to engage them regularly. Once most of the kids have a solution, ask a few to present their solution.

3. After reviewing the results, discuss the importance of creating an efficient solution to use the least amount of resources. Explain how, much like this simple puzzle's objective, computers use different algorithms to develop the most efficient code.

4. If possible, show them an example using google maps showing different paths from point A to point B. There will be a shortest path and 1 or 2 time efficient paths.

The solution is included on the next page.



🟨 = City, no fuel cost

▩ = Road, no fuel cost

🟩 = Forest, impassable

🟥 = Stoplight, 1 gallon

⬜ = Snow, 3 gallons

🟧 = Bridge, no fuel cost

Closing Thoughts:

Once the students have finished the activities deliver a closing talk that gives examples of how algorithms such as the ones in the activities, are used in the real world. Allow the children to ask any questions they may have. Collect the writing utensils from each student and give each of them a copy of a second handout, found in Appendices B, which has extra resources for those interested in programming/algorithms.

Schedule:

Introduction	5 minutes
Sorting Activity	5-10 minutes
Break- Switch Activities	2 minutes
Pathing Activity	5-10 minutes
Concluding Discussion	5 minutes

Teacher resources:

A basic introduction to algorithms.
http://courses.cs.vt.edu/csonline/Algorithms/Lessons/

Khan Academy - introduction to different aspects of programming.
https://www.khanacademy.org/computing/computer-programming

A visual programing language for kids designed by MIT.
http://scratch.mit.edu/

A description of The Traveling Salesman problem
http://support.sas.com/documentation/cdl/en/ornoaug/65289/HTML/default/viewer.htm#ornoaug_optnet_details34.htm

Appendices A: Modified Traveling Salesman

Includes the worksheet that is to be handed out to the kids. It is included on the next page.

Rules:

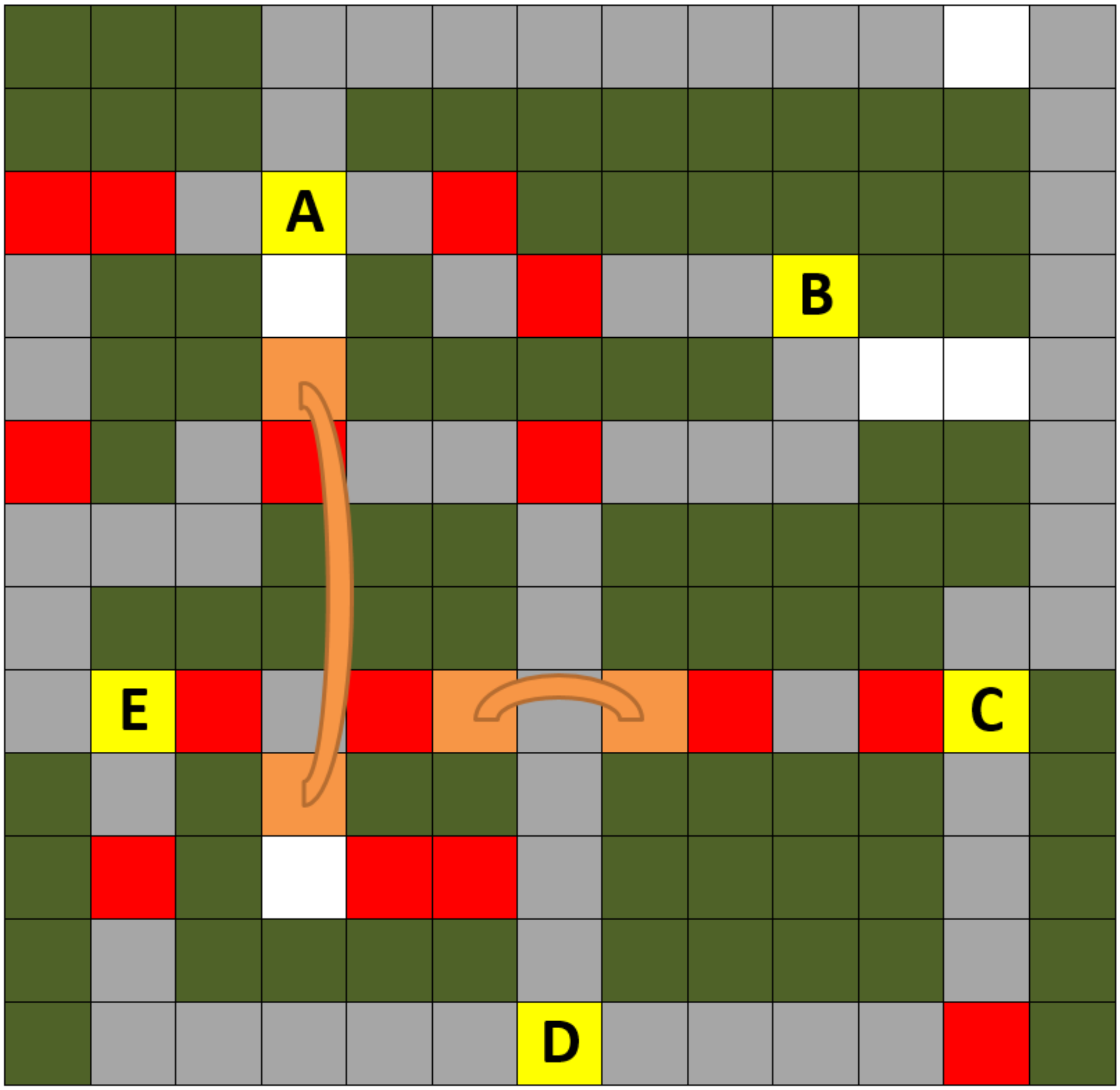
1. *The objective is to find the most efficient path that runs through points A to E by accumulating the least amount of gallons.*

2. *Start at A. Moving can only done in a horizontal or vertical direction.*

3. *Once you start on a bridge, you must continue until the bridge has ended*

4. *Each square can be used no more than once, but you do not have to use all of the squares.*

(Hint: Write down the costs of the individual routes. For example A -> B costs 2 gallons.)



🟩	🟩	🟩	▒	▒	▒	▒	▒	▒	▒	▒	⬜	▒
🟩	🟩	🟩	▒	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	▒
🟥	🟥	▒	A	▒	🟥	🟩	🟩	🟩	🟩	🟩	🟩	▒
▒	🟩	🟩	⬜	🟩	▒	🟥	▒	▒	B	🟩	🟩	▒
▒	🟩	🟩	🟧	🟩	🟩	🟩	🟩	🟩	▒	⬜	⬜	▒
🟥	🟩	▒	🟥	▒	▒	🟥	▒	▒	▒	🟩	🟩	▒
▒	▒	▒	🟩	🟩	🟩	▒	🟩	🟩	🟩	🟩	🟩	▒
▒	🟩	🟩	🟩	🟩	🟩	▒	🟩	🟩	🟩	🟩	▒	▒
▒	E	🟥	▒	🟥	🟧	▒	🟧	🟥	▒	🟥	C	🟩
🟩	▒	🟩	🟧	🟩	🟩	▒	🟩	🟩	🟩	🟩	▒	🟩
🟩	🟥	🟩	⬜	🟥	🟥	▒	🟩	🟩	🟩	🟩	▒	🟩
🟩	▒	🟩	🟩	🟩	🟩	▒	🟩	🟩	🟩	🟩	▒	🟩
🟩	▒	▒	▒	▒	▒	D	▒	▒	▒	▒	🟥	🟩

🟨	= City, no fuel cost	🟥	= Stoplight, 1 gallon
▒	= Road, no fuel cost	⬜	= Snow, 3 gallons
🟩	= Forest, impassable	🟧	= Bridge, no fuel cost

Appendices B: Sources for kids

The sources listed below are additional online resources for interested students.

http://www.thekidzpage.com/learninggames/logic_games/index.html
This website has a variety of logic puzzles for kids.

http://www.squiglysplayhouse.com/PencilPuzzles/
This website has a greater variety of puzzles.

http://courses.cs.vt.edu/csonline/Algorithms/Lessons/
A basic introduction to algorithms.

https://www.khanacademy.org/computing/computer-programming
Khan Academy - introduction to different aspects of programming.