Objectives Students will be able to…

* **Compare and contrast** the different search algorithms.

Assessments Students will...

* **Complete** some short-answer questions.

Homework Students will...

* **Read** HW 13.1 “Sorting.”
* **Complete** self-check questions #4-6 and exercises #1-3

# Materials & Prep

* **Projector and computer**
* **Whiteboard and** **markers**
* **Classroom copies** of WS 7.1
* **CS Unplugged Activity:** Searching Algorithms (<http://csunplugged.org/searching-algorithms/>)
* **Classroom copies** of Battleship 1A, 1A’, 1B, 1B’, 2A, 2A’, 2B, 2B’
  + - * Included in the CS Unplugged activity
    - **10-15 slips of paper** with different integers printed on them (1 integer per paper)
    - **Individually wrapped small candies**
    - **Student pair assignments**

# Pacing Guide

|  |  |
| --- | --- |
| Section | Total Time |
| Bell-work and attendance | 5min |
| Intro & demonstration | 10min |
| Student activity 1: Battleship Linear Searching | 15min |
| Student activity 2: Battleship Binary Searching | 15min |
| Worksheet completion/whole group discussion | 10min |

# Procedure

*Hook your students by placing the bowl of candy somewhere visible. Before the introductory lecture, announce that today’s class is a “game day,” and students will spend their time investigating computer algorithms by playing Battleship.*

## Bell-work and Attendance [5 minutes]

## Intro & Demonstration [10 minutes]

1. Begin with a lecture/discussion about search algorithms.

2. Computers are useful because they can manage large collections of data quickly and easily. Ask students to give some examples about how large collections of information are managed by computers.

* + - * Examples: Data about items for sale are accessed as bar codes, schools store data about students, which can be accessed by student name, ID number, or grade level, weathermen store historical and current data about atmospheric conditions, etc.

3. These data aren’t manageable unless we are able to search for a particular data point (datum).

* + - * Example: When you search the internet, you’re searching for a single keyword (or phrase) called a “search key” within a particular webpage (or set of webpages).

4. Even though computers work very quickly, when we deal with searching large datasets, we need to use algorithms that are quick to use. A difference of a second or two is actually quite a lot when you think about how many times a day we use searches.

* + - * Using your phone, the class clock, a watch, stopwatch, or your computer, demonstrate for the class what 3 seconds feels like. Ask students to imagine each websearch taking that long.
      * Demonstrate 10 seconds, and ask students to imagine what would happen in a grocery store if each item scanned took 10 seconds for the price look-up. Ask for estimates on how long it would take a single family to check out groceries for the week, and have students offer predictions as to how this would affect business and consumer experience in the store.

5. When we decide as program designers which searching or sorting algorithms to use, we factor in:

1. The size of the data array

2. The space efficiency of the algorithm (how much memory it uses)

3. Run-time efficiency (how fast it executes)

6. **DEMONSTRATION:** Using the CS Unplugged guide “Introductory Activity,” get your students thinking about the process of and relationship between searching and sorting data. Use the introductory activity to introduce the Battleship games.

## Student Activity 1: Battleship Linear Searching [15 minutes]

1. On the projector or the board, review the rules for Battleships – A Linear Searching Game from the CS Unplugged activity. Distribute sheets 1A and 1B to student pairs (face down so students don’t see each other’s papers).

2. Distribute WS 7.1 so students can answer questions as they play the Battleship games.

3. Give students ~15 minutes to play the Battleship game and answer the corresponding questions on their worksheets. Students that complete the game with enough time to do a second round should receive 1A’ and 1B’. Be sure to pace your students by announcing 5 minutes before transition.

## Student Activity 2: Battleship Binary Searching [15 minutes]

1. Using the CS Unplugged guide for Battleships – A Binary Searching Game, explain the updated rules for this game. Distribute sheets 2A and 2B to student pairs (face down).

2. Remind students to answer the questions on their worksheets.

3. Give students ~15 minutes to complete the game and answer their worksheets, then call the class together for a whole group discussion of their answers. Students that complete the game with enough time to do a second round should receive 2A’ and 2B’.

## Worksheet Completion/Whole-Group Discussion [10 minutes]

1. Discuss the worksheet questions as a class, assessing student understanding and re-teaching as needed.

# Accommodation and Differentiation

To ensure that students understand the assignment, read the questions on WS 7.1 before students begin the activity.

Help students with metacognition by checking in with student pairs during the activities. Ask them to explain their decision making process to you, and if they are having trouble articulating their algorithms, ask them to explain one decision at a time.

Whenever possible, you should encourage students to do 2 rounds of Battleship for each search algorithm. This will allow students to track their own learning.

If your students are advancing through the course quickly and easily, you can augment this unit by having students write code to implement sequential or binary search. Assign further reading in the textbook (the latter ½ of Chapter 13), and discuss with students how they can implement code that operate like the processes explored during class.