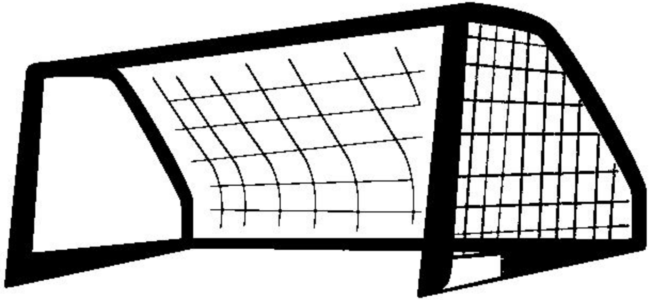
Introduction to Sorting Algorithms

Goal for Today

Today we'll be talking about what a **sorting algorithm** is and why it's so important. We'll also try coding some ourselves.



By the end of the lesson, students will be able to identify and write simple sorting algorithms using iteration, selection, and sequencing.

Warm Up 1

Line up in the order of your birthdays starting with Jan 1 and ending with Dec 31….NO TALKING

Metacognition: So today's theme is sorting. Think about techniques you're using to help sort.

Warm Up 2

Team Blue vs Team Grey

Line up in the order of your birthdays starting with Jan 1 and ending with Dec 31.

This time, utilize any lessons from the first round and try to be the faster group!

Metacognition: So today’s theme is sorting. Think about techniques you’re using to help sort.

Discussion

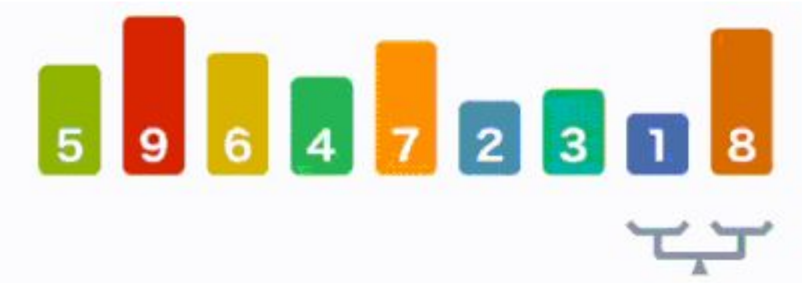
What kind of techniques worked well? What did not?

Selection Sort

- Look through the array for the **minimum** and swap to the front of the array.
- Continue until you reach the end of the array and you have nothing to swap!
- Code-along segment.
- We’ll be coding least to greatest.
    - How could we sort greatest to least?

Bubble Sort

- Start from 1 end of the array and compare its neighbor. If smaller, **swap** it down 1 index. Repeat!
- Code-along segment
  - We will code so that our larger numbers “bubble up” from the beginning.
  - How could we code the gif instead?



Exercise/Assignment

Required:

1. Which sort is better? Come up with 1 pro and 1 con of each type of sort we went over today.
2. What would happen with larger data sets?
3. Pick 1:
   a. Easy: Code an algorithm that looks for the max of an array instead of the minimum
   b. Medium: Code a selection sort algorithm that will sort something from greatest to least.
   c. Hard: Code a bubble sort algorithm that will work as the gif shows--starting from the end of the array and bubbling to the front.