

Assignment 7 - Theory Primer

Problem 1. We saw how `searching` an array could be greatly sped up if the array is sorted. Sorting an array allows us to do many operations faster.

a) Implement `public int findMinSorted(int[] sortedArr)` that finds the minimum of a sorted array. In the worst case, how many steps does your algorithm take? What kind of algorithm is it in Big O Notation? How does this compare to finding the minimum of an unsorted array?

b) Write `public static boolean hasDuplicates(int[] arr)` which returns whether any array has any duplicates.

To do this, you can start with the first element, and check if it is equal to any of the other elements of the array. Then check the second element, and compare it to the rest of the array, etc. This will require a double for loop like `selection sort`.

What is the running time of your algorithm in Big O Notation?

c) Write `public static boolean hasDuplicatesSorted(int[] sortedArr)` which returns whether a *sorted* array has any duplicates. How does its Big O compare to the unsorted version?

Hint: If there are duplicates in the array, where are they in relation to one another? Do we need a double for loop now?