

# Assignment 5

Write documentation of your runtime values and explain the runtime of each method and the big oh of the following methods

$O(\log N)$  basically means time goes up linearly while then goes up exponentially. So if it takes 1 second to compute 10 elements, it will take 2 seconds to compute 100 elements, 3 seconds to compute 1000 elements, and so on. It is  $O(\log n)$  when we do divide and conquer type of algorithms e.g binary search.

The way that quicksort uses divide-and-conquer is a little different from how merge sort does. In merge sort, the divide step does hardly anything, and all the real work happens in the combining step. Quicksort is the opposite: all the real work happens in the divide step. In fact, the combining step in quicksort does absolutely nothing.

Which implementation is the fastest?

The implementation that is the fastest is Mergesort and Quicksort because of the Big O of  $O(n \log n)$ . The runtime of Quicksort and Mergesort have the same big Oh time complexity which means that each time you run this program there are times when Mergesort will be faster and quicksort will be faster.

The Bubblesort is  $O(n^2)$  that is why it is the slowest sorting algorithm because it has to go through the array multiple times.

The runtime of Quicksort and Mergesort have the same big Oh time complexity which means that each time you run this program there are times when Mergesort will be faster and quicksort will be faster.

