

PROGRAM 2 REPORT BY MD ABUBAKKAR
CMPS 5243- ALGORITHM ANALYSIS
SPRING 2022
Dr. TINA JOHNSON

For Program 2, we have implemented 3 different sorting algorithms on arrays of size 5000. One sort from each runtime category: $O(n^2)$, $O(n \log n)$, $O(n)$. I chose Counting Sort for $O(n)$, Quick Sort for $O(n \log n)$, and Selection Sort for $O(n^2)$. The purpose of this program is to understand the difference between runtime complexity between mentioned algorithms.

We know the quicksort is infamously a divide and conquer approach with recurrence relation. The runtime analysis is complicated because the QuickSort algorithm actually calls itself twice! (the Recursion is coming!). Selection sort on the other hand, was selecting biggest and smallest number. We have increased the array size and it was taking too much time. So, clearly Selection sort isn't the best choice for bigger array. Counting sort, lastly, actually has linear time complexity but we can't say that it's the best algorithm because the space complexity is quite high and it is only suitable to use in a scenario where input array element range is close to the size of the array. Since for all sorting algorithm, we filled out the array with random number, we had different results everytime we ran a sort algorithm.

I believe the assignment was a good practice to learn standard sorting algorithm and how they behave and which one is preferable in which situation. Geeksforgeeks and Stackoverflow helped me understand some timer library of c++ which was crucial for the project.