Data Structures and Algorithms

Homework # 3

Eren Erdogan

Question 1:

For this question we were asked to match the columns with the sorting algorithm that generated the partially sorted column.

The results I got can be seen below.

Column #	Sorting Algorithm
2	Quicksort
3	Top-Down Mergesort
4	Insertion Sort
5	Selection Sort
6	Bottom Up Mergesort
7	Shell Sort

No trace was asked in this question and I emailed Professor Ra to make sure we did not need to provide a trace.

Question 2:

Please run code, refer to instructions on how to run on Sakai. Also please view testing results txt file in folder to see test input cases and results.

For this questions first part we were asked to compare the performance of Quicksort With Median-of-3 against Top-Down Mergesort.

In theory we know that Quicksort in general is faster than Mergesort when dealing with integers. We also know that the Median-of-3 addition to standard quicksort is designed to improve it's performance by selecting three random numbers and taking the median of those and using that use the pivot. This is to try and select a better pivot than if it were randomly selected with a shuffle.

We can see the testing results we got below.

Quick-Sort Median-of-3 vs Top-Down Mergesort		
	Algorithm Run Time (Seconds)	
Input Size (N)	Quick-Sort w/ Median-of-3	Top-Down Mergesort
10	0.001	0
1,000	0.002	0.005
100,000	0.018	0.024
1,000,000	0.161	0.184
10,000,000	1.704	2.253

We see that Quicksort Median-of-3 is indeed faster than Top-Down Mergesort.

After we update both algorithms with the insertion sort cutoff of 10 we can see that there is an improved performance overall. The data for this can be seen in the table below.

Quick-Sort Median-of-3 vs Top-Down Mergesort with Insertion Sort Cutoff of 10		
	Algorithm Run Time (Seconds)	
Input Size (N)	**Quick-Sort w/ Median-of-3**	**Top-Down Mergesort**
10	0	0
1,000	0.001	0.001
100,000	0.027	0.036
1,000,000	0.15	0.155
10,000,000	1.5	1.866

We see that both algorithms received a performance boost with the insertion sort cutoff of 10.

Question 3:

Please run code, refer to instructions on how to run on Sakai. Also please view testing results txt file in folder to see test input cases and results.

For the search of rank(15), 14097 was returned.

For the search of select(9), 1 was returned.