CS 201 HW5

Can Yılankıran S011483

1)Experimental Setup

The sizes of test files are

- 5000
- 6000
- 7000
- 8000
- 9000

Due to the small size of the test files, each test file has 4 randomly generated samples and the average of those 4 samples are shown in this report.

There is 4 Algorithms in this test which are ;

1. SortAll which uses insertion sort.Worst case runing time O(n^2)
2. SortK which also uses insertion sort but it only sorts the only first k elements then checks next element which is k+1. It does operations according to k+1 is small or bigger.Worst case runing time O(n^2)
3. SortHeap which uses minheap to sort numbers.Worst case runing time O(nlogn)
4. SortQuick which uses QuickSelect and QuickSort.Worst case runing time O(n)

2)Results

All results are in seconds.

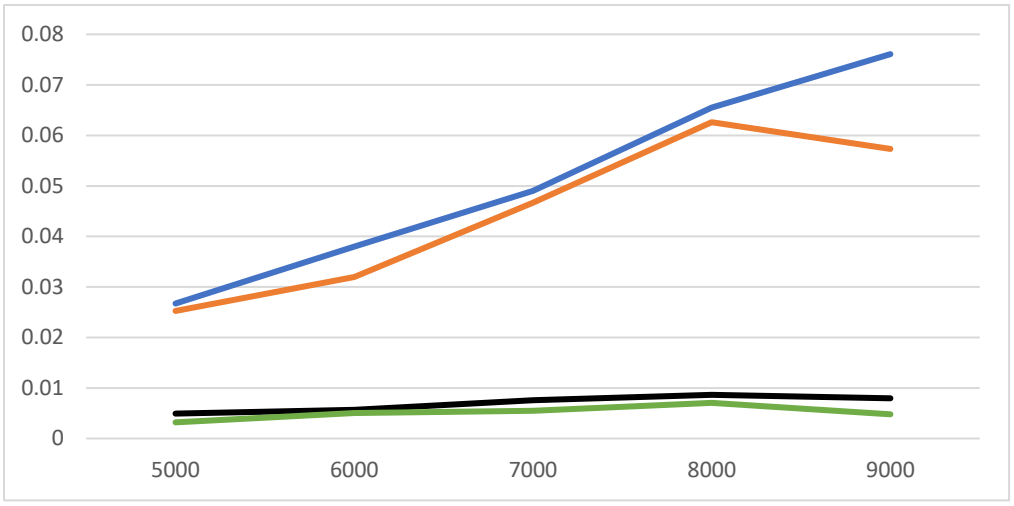
Sizes and algorithms	5000	6000	7000	8000	9000
SortAll	0.02671775	0.03799875	0.04902962	0.06546694	0.07607932
SortK	0.02524814	0.03197487	0.04665452	0.06260452	0.05732758
SortHeap	0.00493325	0.00569173	0.00760518	0.00864266	0.00795237
SortQuick	0.00319871	0.00502614	0.00547921	0.00705453	0.00478932

There is a graph comparing sorting times

In the graph blue line is SortAll algorithm,

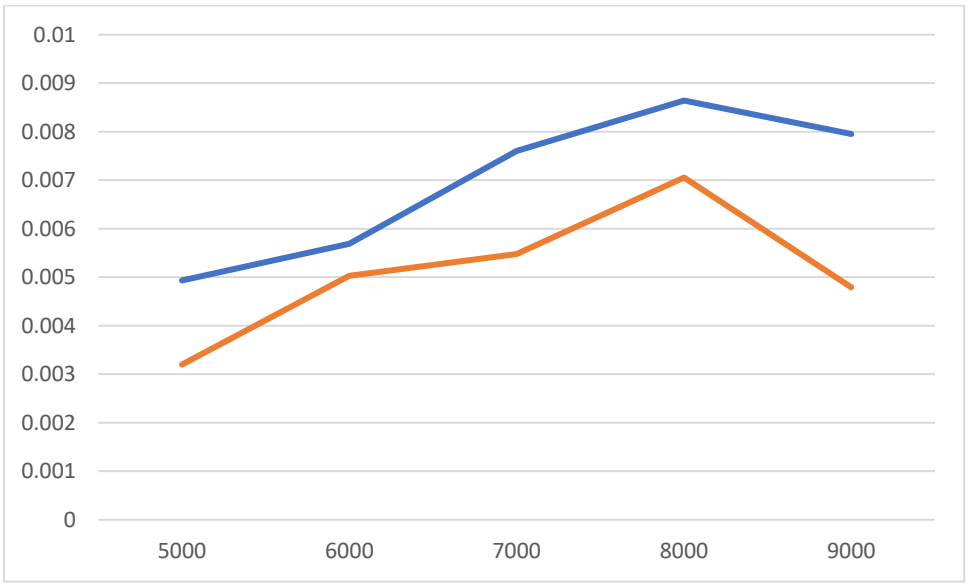
Orange line is SortK algorithm,

Black line is SortHeap algorithm and Green line is SortQuick algorithm



The next graph is only shows the difference between SortHeap algorithm and QuickSort algorithm

Blue line indicates SortHeap algorithm and orange line indicates the QuickSort algorithm



3)Discussion

Firstly, SortK is faster than SortAll.These two algorithm uses same sort mechanism but SortK is more powerful sorting option as seen in test cases.The reason behind that SortK sorts until kth element then starts to check k+1 bigger or not. This operation lowers the cost.

As expected QuickSort is the most powerful sorting algortihm between these 4.The difference is not huge between SortHeap and QuickSort as seen in the first graph but second graph shows QuickSort overpowers SortHeap.

I choosed small input sizes in order to finish all runs.I tried 5 million sized input to see that.5M input did not completed for a long time so i decided to choose smaller input sizes.