Algorithm Programming Assignment #1 **Report**

B07901021 潘世軒

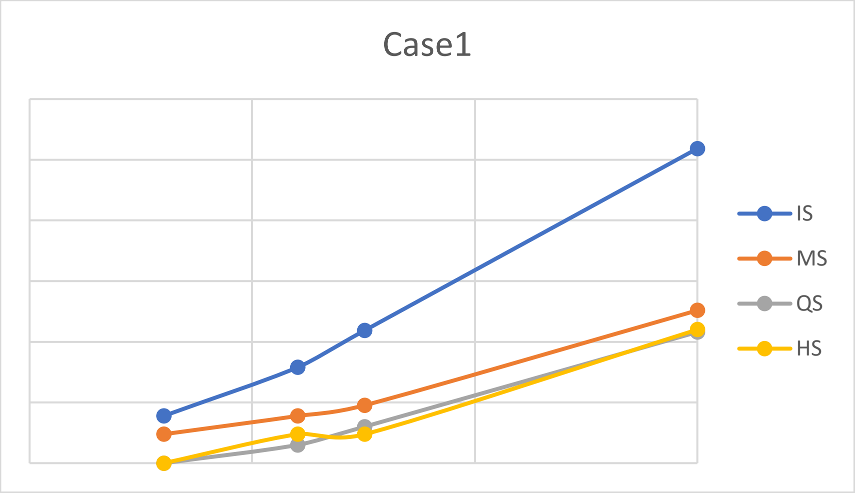
1. Comparison of running time of four versions of different input sizes

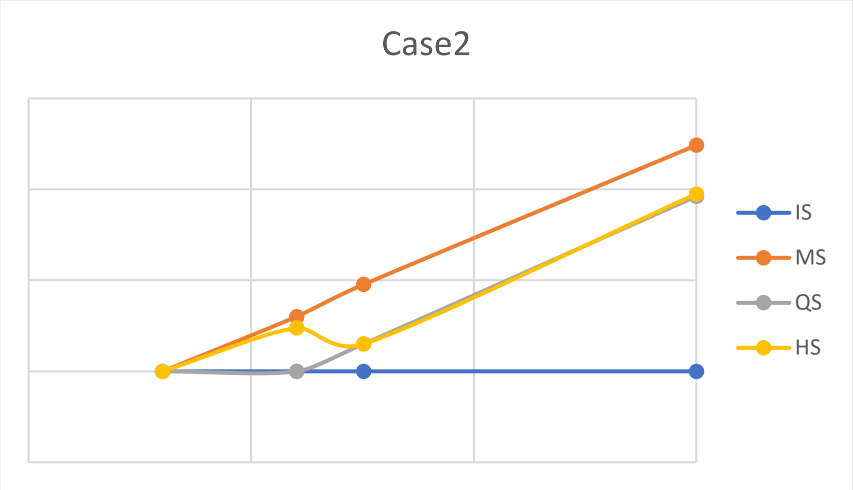
Inputsize	IS		MS		QS(randomized partition)		HS	
	CPU time (ms)	Memory (kB)	CPU time (ms)	Memory (kB)	CPU time (ms)	Memory (kB)	CPU time (ms)	Memory (kB)
4000.case.2	0	12424	1	12424	1	12424	1	12424
4000.case.3	9.999	12424	2	12424	0	12424	1	12424
4000.case.1	5.999	12424	3	12424	1	12424	0	12424
16000.case.2	0	12572	3.999	12572	1	12572	3	12572
16000.case.3	79.988	12572	4	12572	1	12572	2	12572
16000.case.1	37.994	12572	6	12572	2	12572	3	12572
32000.case.2	0	12572	8.998	12760	1.999	12572	2	12572
32000.case.3	307.953	12572	8.998	12760	2.999	12572	2	12572
32000.case.1	153.977	12572	8.998	12760	4	12572	2.999	12572
1000000.case.2	1	18592	305.952	22684	83.987	18592	88.986	18592
1000000.case.3	306709	18592	291.957	22684	83.987	18592	87.987	18592
1000000.case.1	153279	18592	329.95	22684	144.978	18592	158.976	18592

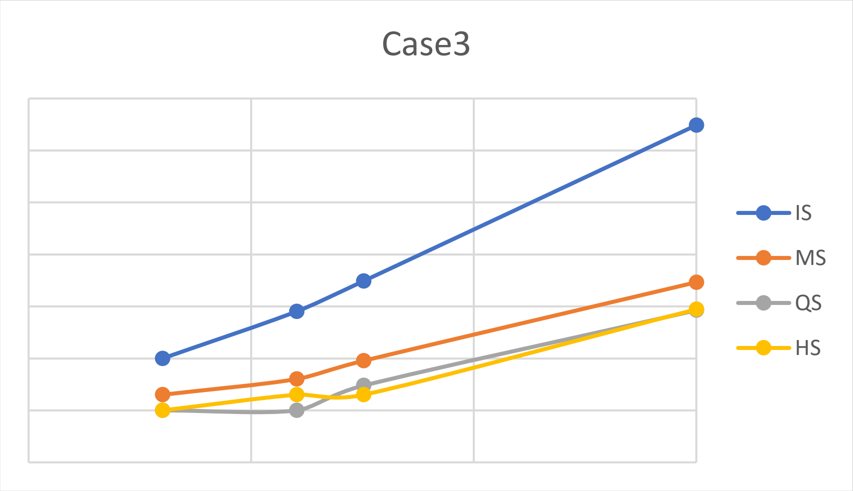
Use EDAU15 server in EDA Union Workstation

EDAU15	CentOS 6.8	Intel Xeon X5680 @ 3.33 GHz	40 GB	gcc 5.4.0

2. Figures showing the growth of running time (function of input size)







(1) The x-y axis is in log scale
(2) For the convenience to plot graph, those case with running time = 0ms are modified to 1ms (almost the same).

3. Analysis:

(1) For case1 (unsorted array), we can find that the growth of running time of MS, QS, HS are similar.(Theoretically, it's O(nlogn)). However, the growth of running time of IS is faster, obviously.(Theoretically, it's O(n^2)).

(2) For case2 (sorted array), IS is the fastest, that's because insert time is O(1), and that the total time complexity become O(n)(which is the best case of IS).

(3) Result of case3 is similar to that of case1.

(4) Overall, The order of time required should be

$$\text{IS} \gg \text{MS} > \text{HS} \approx \text{QS}$$

$$O(n^2) \qquad O(n\log n)$$

4. In PA1, there isn't any specific data structure used, I just followed pseudo code taught in the class to finish all TODOs. However, I find that if I have to sort an array, I would use "QuickSort", because it's the fatest and that QS is easier to understand for me comparing to "HeapSort".