PA1 Report

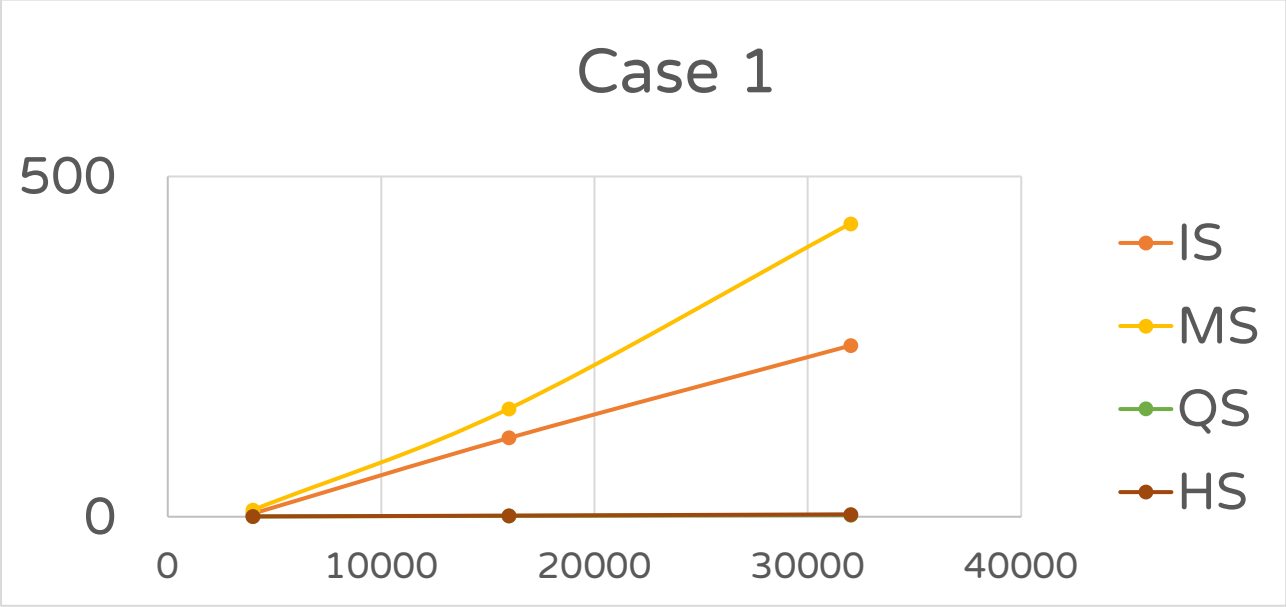
B07505024 劉厚均

Input size	IS		MS		QS		HS	
	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)
5.case1	0.046	14016	0.049	14016	0.047	14016	0.071	14016
4000.case1	4.687	14016	10.106	14152	0.462	14016	0.397	14016
4000.case2	0.088	14016	12.036	14152	19.682	14076	0.396	14016
4000.case3	14.68	14016	13.322	14152	16.389	14016	0.294	14016
16000.case1	115.779	14168	158.905	14168	1.272	14168	1.64	14168
16000.case2	0.108	14168	116.251	14168	215.795	14792	1.434	14168
16000.case3	128.383	14168	120.136	14168	172.919	14420	1.505	14168
32000.case1	251.475	14300	430.899	14300	2.686	14300	3.497	14300
32000.case2	0.137	14300	428.043	14300	788.253	15616	2.601	14300
32000.case3	477.631	14300	425.245	14300	649.295	14856	2.997	14300
1000000.case1	227100	20256	459979	22112	100.16	20256	224.496	20256
1000000.case2	0.857	20256	463094	22112	792316	64952	107.717	20256
1000000.case3	459978	20256	471852	22112	465249	35360	105.677	20256

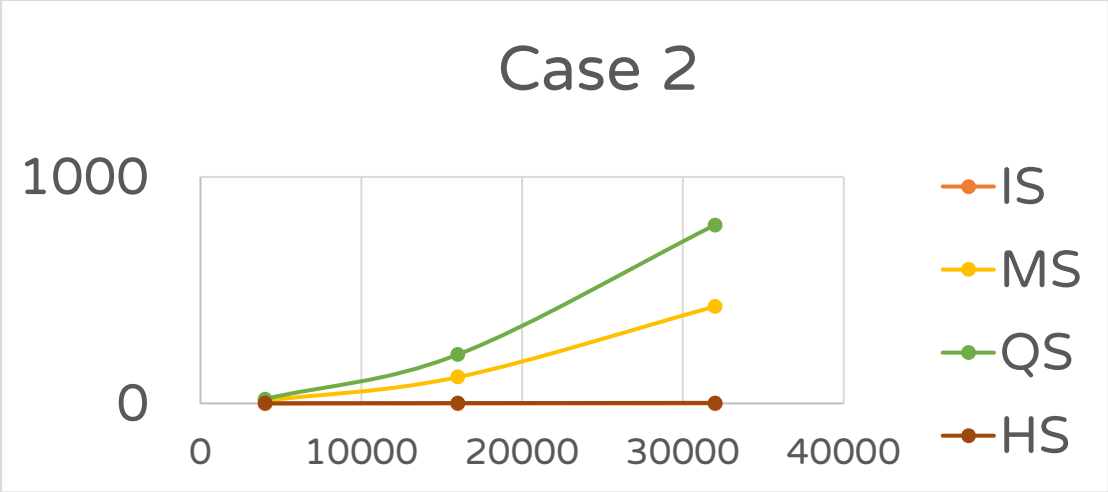
case2 is already sorted in ascending way, and case3 is already sorted in descending way. Which would affect some of the algorithms a lot.

For insertion sort, it performs great in case2 and terrible in case1 and case3.
For merge sort, it performs equally in each case.
For quick sort, case1 performs the best, because the items are equally separated.
For heap sort, it performs equally in each case.

Curve analysis



Case 1 is equally distribution, which is the best case for Quick sort.
In other hand, it is the average case for Insertion sort, which performs $O(n)$. And for Merge sort and Heap sort, the size of inputs don't affect the speed.



Case 2 is a sorted input, which is the best case for Insertion sort, which performs $O(n)$.
In other hand, it is the worst case for Quick sort, which performs $O(n^2)$.

Case 3 is a reversed sorted input, which is the worst case for Quick sort and Insertion sort, which are both performing $O(n^2)$.

After comparing three cases, we can see that Heap sort is the steadiest algorithm, which performs $O(nlogn)$ under any situation. Besides, there is no significant difference between Heap sort and Insertion sort under the best case of the later one, $O(n)$.
So, using Heap sort in uncertain inputs may be a good strategy.