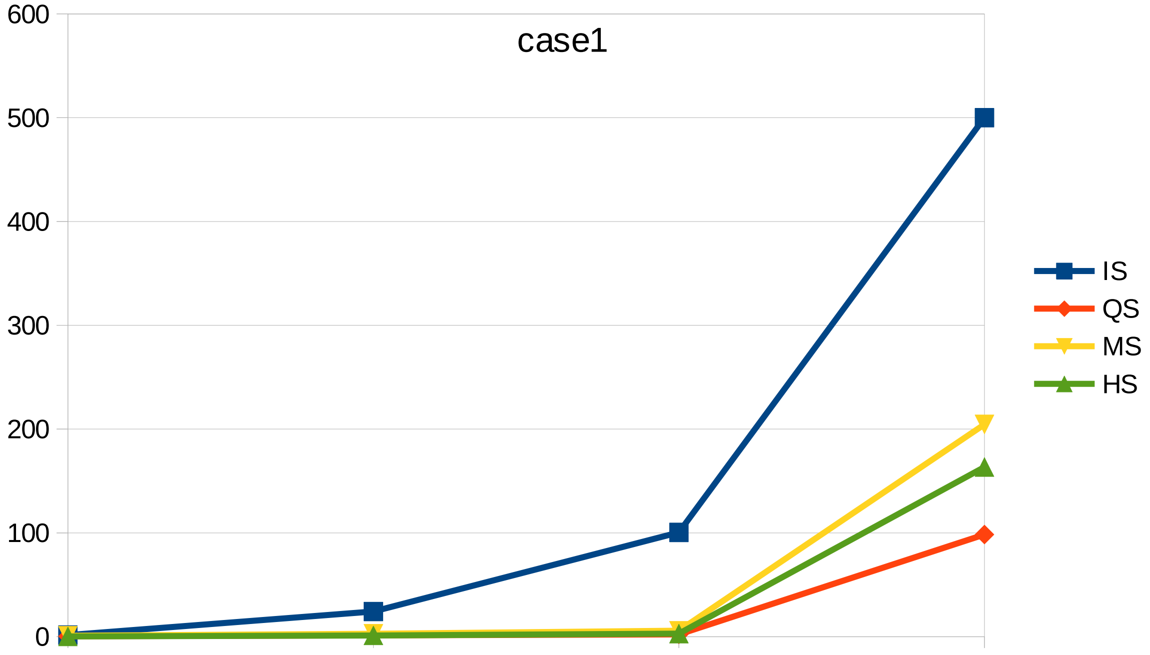
algorithm_HW1 report
b07901088 陳秉嘉

(1) comparison table

Input size	IS		QS		MS		HS	
(ms)/(KB)	CPU time	Memory	CPU time	Memory	CPU time	Memory	CPU time	Memory
4000.case2	0.141	5892	4.793	5892	0.455	6028	0.22	5892
4000.case3	3.268	5892	4.557	5892	0.455	6028	0.25	5892
4000.case1	1.561	5892	0.308	5892	0.919	6028	0.26	5892
16000.case2	0.051	6044	77.44	6044	1.712	6044	1.14	6044
16000.case3	49.463	6044	70.295	6296	1.755	6044	1.05	6044
16000.case1	24.192	6044	1.27	6044	2.817	6044	1.21	6044
32000.case2	0.108	6176	289.5	6176	3.602	6304	2.52	6176
32000.case3	191.133	6176	244.347	6728	3.561	6304	2.26	6176
32000.case1	100.451	6176	2.255	6176	5.749	6304	2.97	6176
1000000.case2	0.868	12132	242842	12132	123.196	16224	94.36	12132
1000000.case3	229578	12132	190804	23984	113.301	16224	92.02	12132
1000000.case1	99496	12132	98.444	12132	204.545	16224	163.42	12132

(2) runtime comparison chart for each case

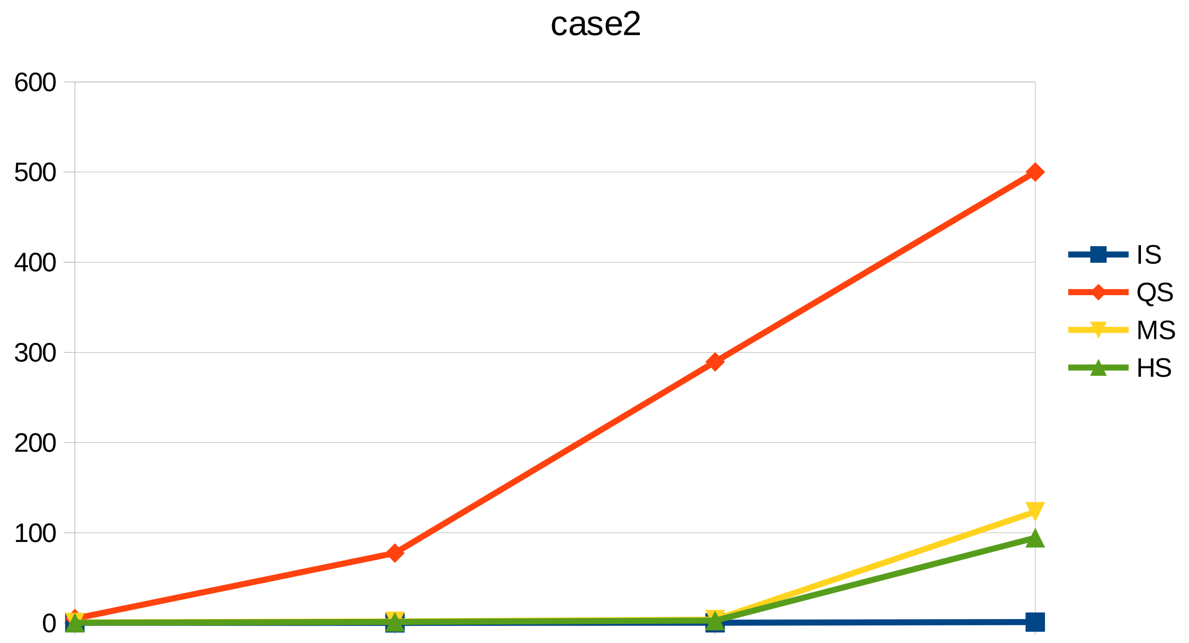
a. case1



For the case1, which is the average case.
The runtime comparison is: Fast => Slow
Quick sort => Heap sort => Merge sort => Insertion sort

We see the result is similar to our expectation because QS、MS、HS are Θ(nlgn) and IS is Θ(n^2) for the average case.

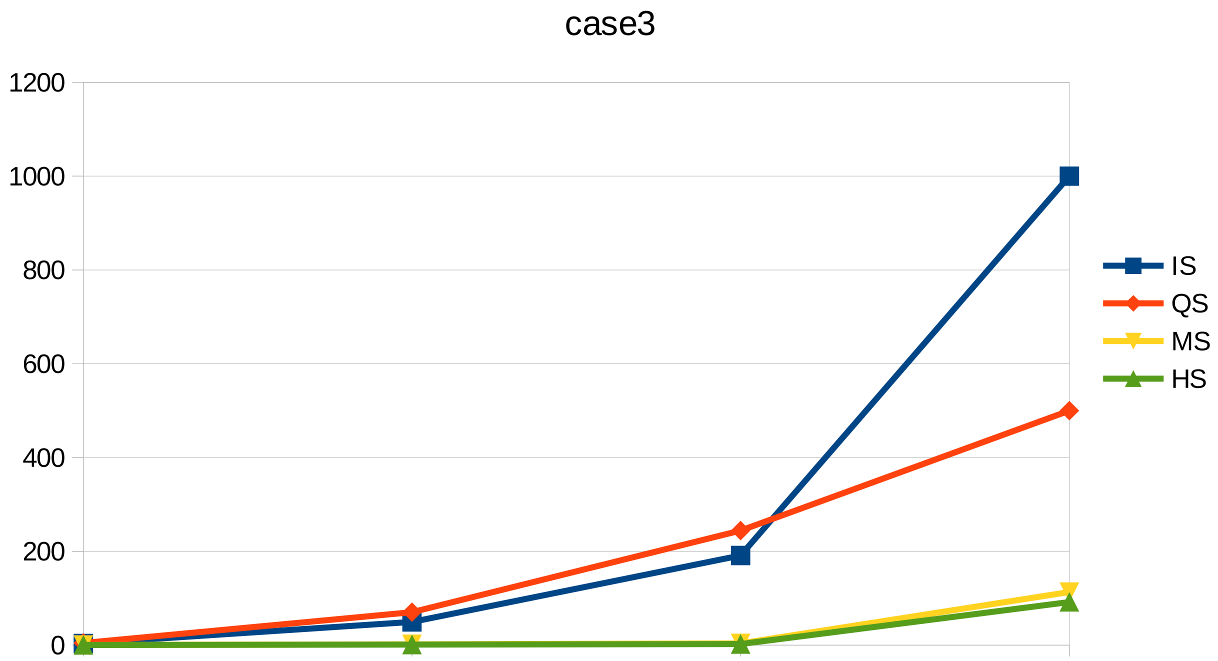
b. case2



For the case2, which is the best case.
The runtime comparison is : Fast => Slow.
Insertion sort => Heap sort => Merge sort => Quick sort

We see the result is similar to our expectation IS is Θ(n) and MS、HS are Θ(nlgn) but QS is Θ(n^2) for the best case.

c. case3



For the case3, which is the worst case.
The runtime comparison is : Fast => Slow
Heap sort => Merge sort => Quick sort => Insertion sort

We see the result is similar to our expectation because MS、HS are Ɵ(nlgn) and IS、QS are Ɵ(n^2) for the worst case.