

Algorithm PA1  
B07611033 電機三 林書宇

	IS		MS		QS		HS	
input size	CPU time(s)	Memory (KB)	CPU time(s)	Memory (KB)	CPU time(s)	Memory (KB)	CPU time(s)	Memory (KB)
4000.case2	0.000088	5892	0.0000742	5892	0.008665	5964	0.000528	5892
4000.case3	0.007303	5892	0.000026	5892	0.00796	5892	0.000528	5892
4000.case1	0.004771	5892	0.0001404	5892	0.000216	5892	0.002194	5892
16000.case2	0.000109	6044	0.0002933	6044	0.131277	6676	0.001142	6044
16000.case3	0.084595	6044	0.000113	6044	0.11495	6296	0.00113	6044
16000.case1	0.058773	6044	0.0001858	6044	0.000881	6044	0.001417	6044
32000.case2	0.000258	6176	0.000436	6176	0.507285	7492	0.004213	6176
32000.case3	0.31521	6176	0.0001846	6176	0.405732	6728	0.003168	6176
32000.case1	0.163094	6176	0.0003557	6176	0.002104	6176	0.002976	6176
1000000.case2	0.002376	12132	0.0058189	13992	579.82	56840	0.070333	12132
1000000.case3	320.576	12132	0.006226	13992	339.309	27244	0.069212	12132
1000000.case1	156.375	12132	0.0120478	13992	0.067342	12132	0.219952	12132

case1	IS	MS	QS	HS
4000	0.004771	0.0001404	0.000216	0.002194
16000	0.058773	0.0001858	0.000881	0.001417
32000	0.163094	0.0003557	0.002104	0.002976
1000000	156.375	0.006226	0.067342	0.219952

case2	IS	MS	QS	HS
4000	0.000088	0.0000742	0.008665	0.000528
16000	0.000109	0.0002933	0.131277	0.001142
32000	0.000258	0.000436	0.507285	0.004213
1000000	0.002376	0.0058189	579.82	0.070333

case3	IS	MS	QS	HS
4000	0.007303	0.000026	0.00796	0.000528
16000	0.084595	0.000113	0.11495	0.00113
32000	0.31521	0.0001846	0.405732	0.003168
1000000	320.576	0.006226	339.309	0.069212

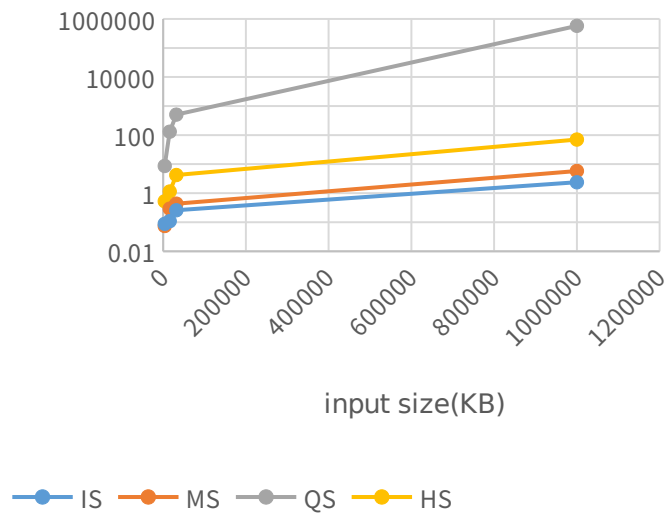
Run Time(ms) (log scale)

case1



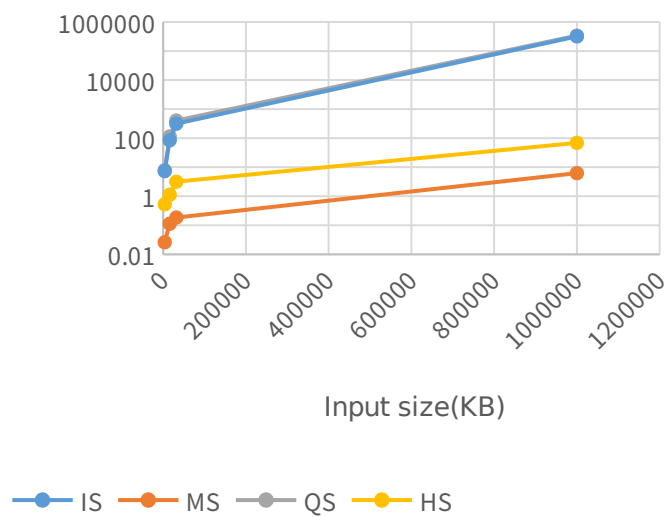
Run time(ms) (log scale)

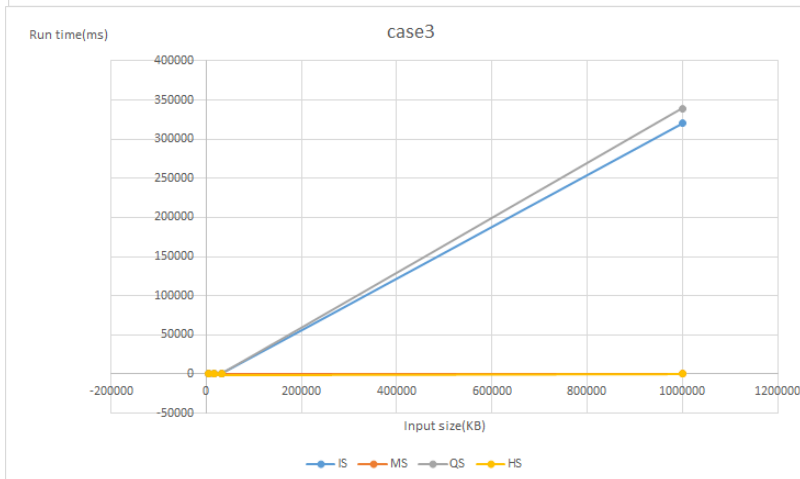
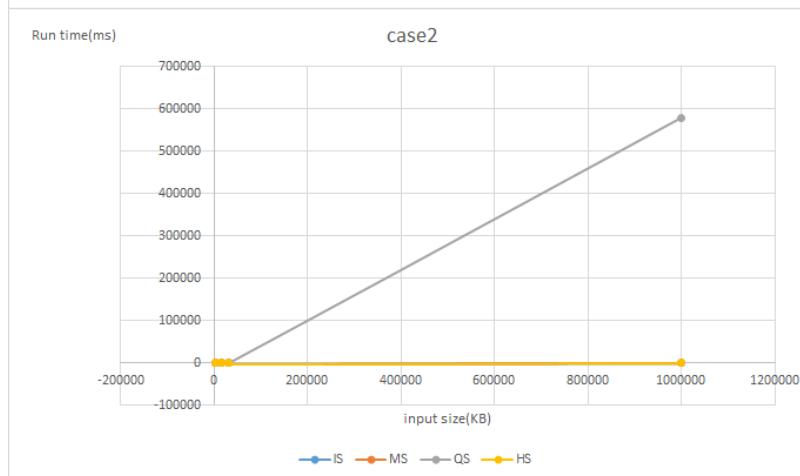
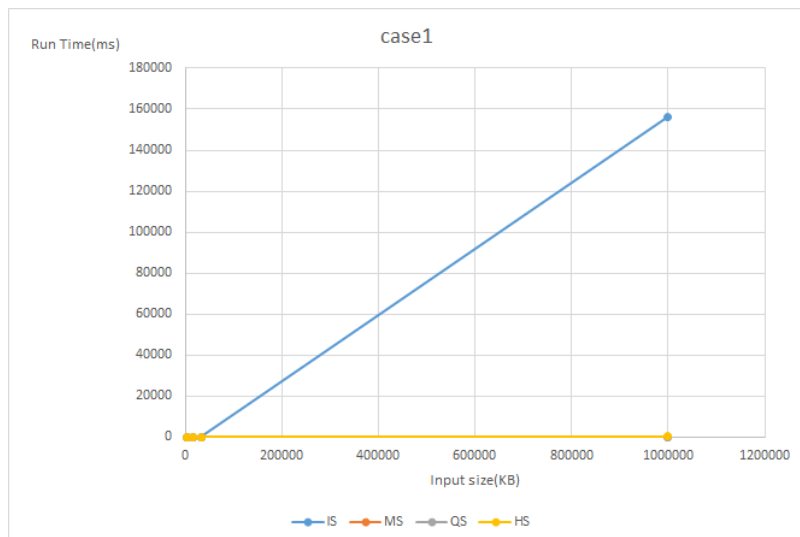
case2



Run time(ms) (log scale)

case3





分析：

( ) case1 (random) : 在 random case 下，insertion sort 有最差的表現，符合他在 average case 中  $\Theta(n^2)$  的性質，雖然說在 input size= 32000 與 input size=1000000 這兩個 case 的 Run time 之間，沒有很明確的跟資料量的平方有關係，但確實與其他種 sort 有著明顯的差異，突顯  $n^2$  是極快的成長速度。

( ) case2(sorted order) : 在 sorted order case 下，quick sort 有最差的表現，與其他三種有明顯的差異，這是由於我在實現 quick sort 的時候，總是將數列的最後一個數設為 pivot，如此一來，他將會進行數量極多的 partition，這是我們所不樂見的，因此 randomized partition 可以解決這個問題。

( ) case3(reversed order) : 在 reversed order case 下, insertion sort & quick sort 有最差的表現, quick sort 的原因已於第二點闡述, insertion sort 的原因則是因為在 reversed order 下, 每次皆要移動大量資料, 造就了 insertion sort 在 worst case(reversed order)下  $O(n^2)$  的時間複雜度。