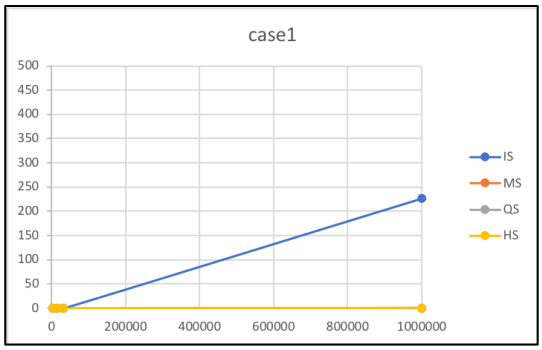
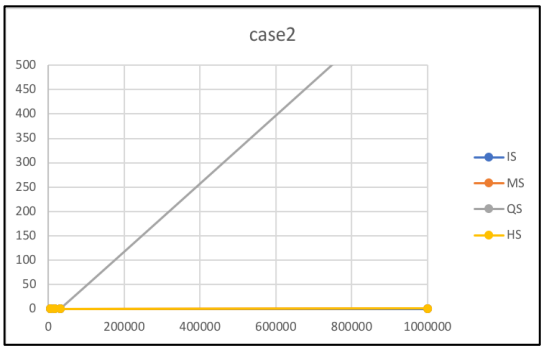
電機三 孟妍 b06901066

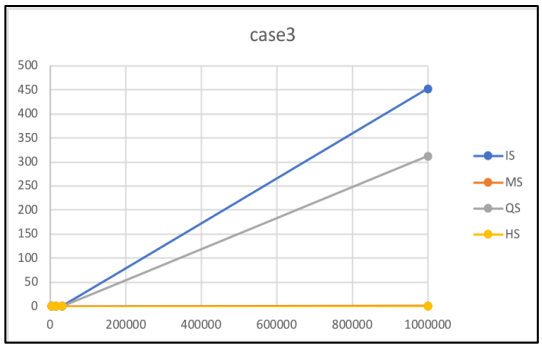
PA1 Report

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)
4000.case2	0	12424	0.002	12424	0.010998	12540	0	12424
4000.case3	0.008999	12424	0.002	12424	0.010999	12440	0	12424
4000.case1	0.003999	12424	0.001	12424	0.001	12424	0	12424
16000.case2	0	12572	0.004	12572	0.154997	13248	0.002	12572
16000.case3	0.116983	12572	0.005	12572	0.114982	12876	0.002	12572
16000.case1	0.056991	12572	0.004	12572	0.002	12572	0.003	12572
32000.case2	0	12572	0.007999	12944	0.615906	13996	0.002	12572
32000.case3	0.461982	12572	0.008998	12944	0.454931	13240	0.002	12572
32000.case1	0.229965	12572	0.008998	12944	0.002	12572	0.002	12572
1000000.case2	0.001	18592	0.26494	28260	676	18952	0.086987	18592
1000000.case3	452.003	18592	0.265959	28260	312	18952	0.085986	18592
1000000.case1	225.968	18592	0.322952	28260	0.086986	18952	0.157976	18592

→ time complexity growth curve







→ result analysis

我把 y 軸範圍都設成 0~500 方便觀察跟比較。

• case1:

case1 是 average case，因此只有 insertion sort 是$O(n^2)$比較慢，其餘的 sorting 都是$O(nlogn)$所以都很快，即使到 1000000 的 case 都還花不到 1ms

• case2:

case2 是由小到大排好的，對 insertion sort 來說是 best case 是 linear time 所以非常快，對 merge sort, heap sort 來說不管什麼情形都是$O(nlogn)$所以一樣也是非常快但也沒有因為原本已經排好了所以更快。但對 quick sort 來說 case2 是 worst case，因為每次取的 pivot 都是從最右邊取，所以 partition 極不平均，用 quick sort 跑 1000000.case2.in 是所有跑的裡面最久的。

• case3:

case3 是由大到小排好的，對 insertion sort 和 quick sort 來說是 worst case，是$O(n^2)$所以很慢，其餘的 sort 就一樣很快。但就 case3 來說，quick sort 還是有比 insertion sort 快一些。

總結來說，如果是小資料的 case，可以直接用 insertion sort 就好比較方便，但如果資料量大，就要考慮用其他三種，merge sort 和 heap sort 不會受到 input 資料分布影響，所以是任何情形都通用的。Quick sort 就必須有條件地使用否則就會很慢，但如果能讓 partition 平均一點就會很快，比較最大的 average case (1000000.case1.in)，quick sort 是比 merge sort, heap sort 更快一些。