

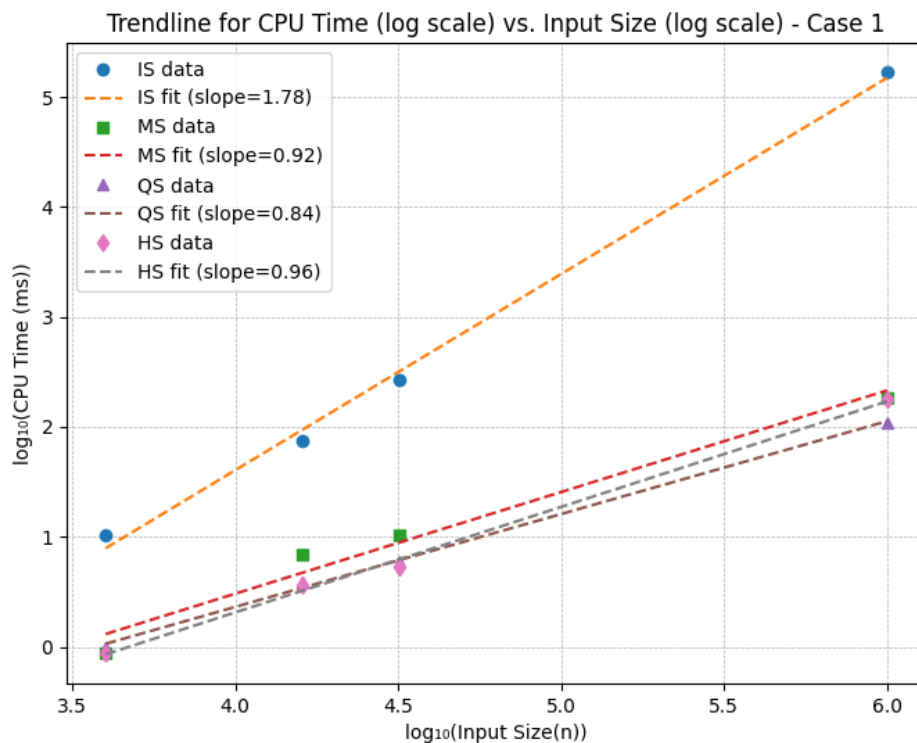
演算法 PA1

電機二 B12901075 賴禹衡

Using EDA union lab machines

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	0.09m	6072	0.893m	6072	23.763m	6136	0.691m	6072
4000.case3	18.019m	6072	0.916m	6072	22.828m	6072	0.735m	6072
4000.case1	10.37m	6072	0.88m	6072	0.981m	6072	0.87m	6072
16000.case2	0.108m	6224	3.489m	6224	171.518m	6848	2.845m	6224
16000.case3	119.447m	6224	3.223m	6224	175.142m	6476	1.022	6224
16000.case1	75.516m	6224	6.956m	6224	3.877m	6224	3.595m	6224
32000.case2	0.248m	6356	5.699m	6356	637.971m	7644	6.149m	6356
32000.case3	445.577m	6356	5.139m	6356	647.068m	6916	4.671m	6356
32000.case1	268.48m	6356	10.46m	6356	6.051m	6356	5.333m	6356
1000000.case2	1.056m	12312	75.286m	14172	232882m	39756	92.001m	12312
1000000.case3	319562m	12312	82.162m	14172	354734m	28288	88.579m	12312
1000000.case1	169497m	12312	183.225m	14172	110.097m	12312	176.588m	12312

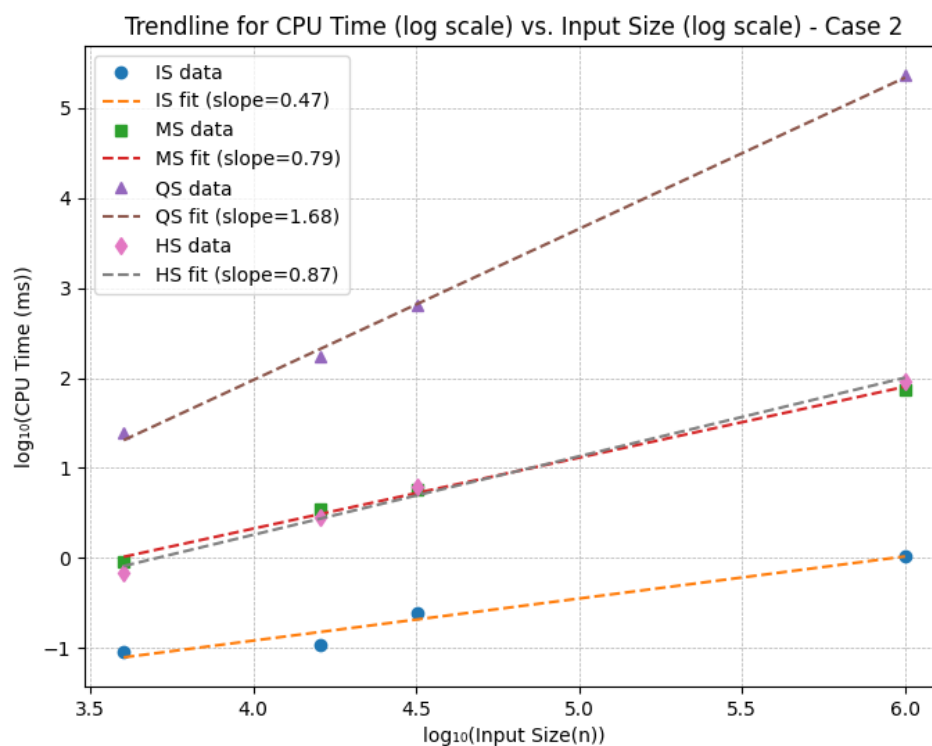
圖一：Table of runtime and memory usage of four sorting algorithms



圖二：Average Case (Case 1)

在 Average Case 下，Insertion Sort 是 $O(n^2)$ ，其他的演算法都會是 $O(n \log n)$ ，如圖，Insertion Sort 的斜率接近 2 (取 $\log n$)，而其他的排序演算法都在 0.8 到 0.9 左右，理論上 $n \log n$ 的斜率應該要是 1~2 之間，但我推測因為本次作業小測資比較多，因此畫出來的漸近線會比 1 還要小。

而此結果與 example 大致相同。

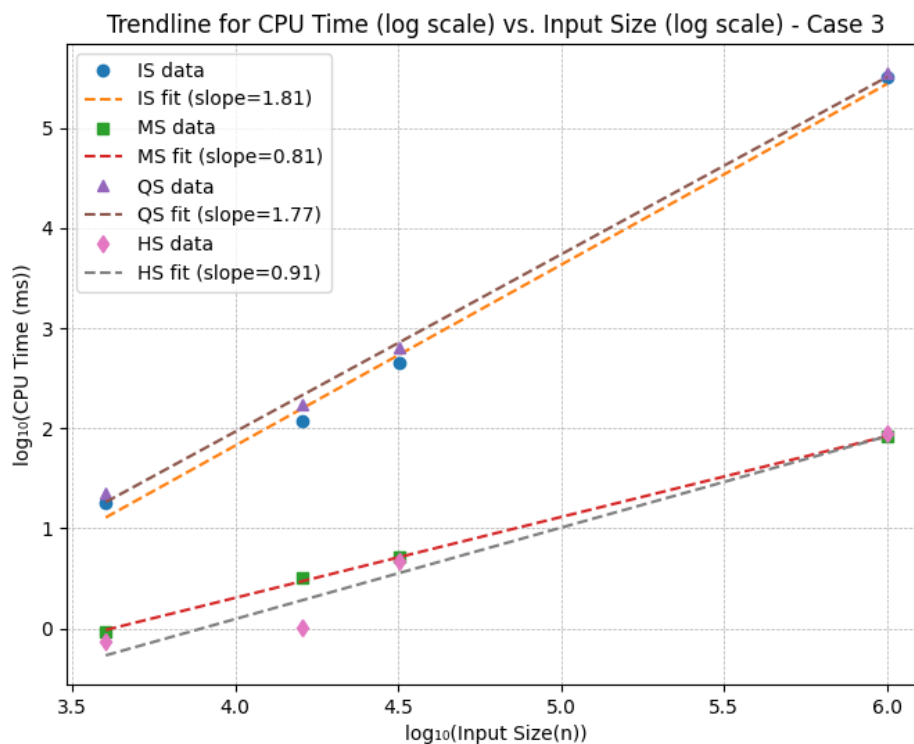


圖三：Best Case (Case 2)

在 best Case 下，Insertion Sort 是 $O(n)$ ，Quick Sort 是 $O(n \log n)$ ，其他的演算法都會是 $O(n \log n)$ ，如圖，Quick Sort 的斜率接近 2（取完 $\log n$ ），而 Merge Sort 與 Heap Sort 排序演算法都在 0.8 到 0.9 左右，Insertion Sort 為 0.5 左右，理論上 $n \log n$ 的斜率應該要是 1~2 之間， n 則要接近 1，但我推測因為本次作業小測資比較多，因此畫出來的漸近線會偏離。

Insertion Sort 在最佳的情況下，當陣列已經基本有序時，每次只需進行極少的移動所以會是 $O(n)$ 。

而此圖雖然看起來跟 example 不太一樣，但應該是 x 軸區間只有 3.5 – 6.0 當尺度拉大就會類似



圖四：Worst Case (Case 3)

在 Worst Case 下，Insertion Sort 和 Quick Sort 是 $O(n^2)$ ，其他的演算法都會是 $O(n \log n)$ ，而這點在圖中可以清楚看到，Insertion Sort 和 Quick Sort 的斜率是接近 2，而其他二種排法都會在 0.8 到 0.9 左右，而原因推測跟上述幾題測資大小問題有關。Insertion Sort 在最壞的情況下，那麼每次插入都需要走過大部分已排序部分，最壞情況的時間複雜度是 $O(n^2)$ Quick Sort 在最壞的情況下，當每次選的 pivot 都位於極端位置（數列已經有序，而每次都選到第一個或最後一個元素作 pivot），就會產生極度不平衡的分割，導致時間複雜度退化為 $O(n^2)$

這張圖雖然跟 example 有點不同，但整體趨勢是符合理論值的（二個 $O(n^2)$ 二個 $O(n \log n)$ ）