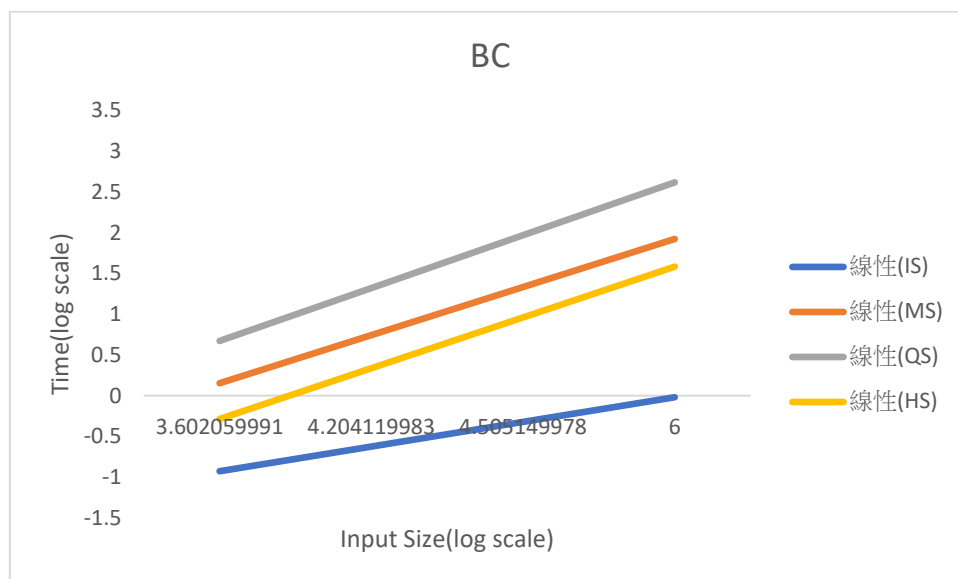
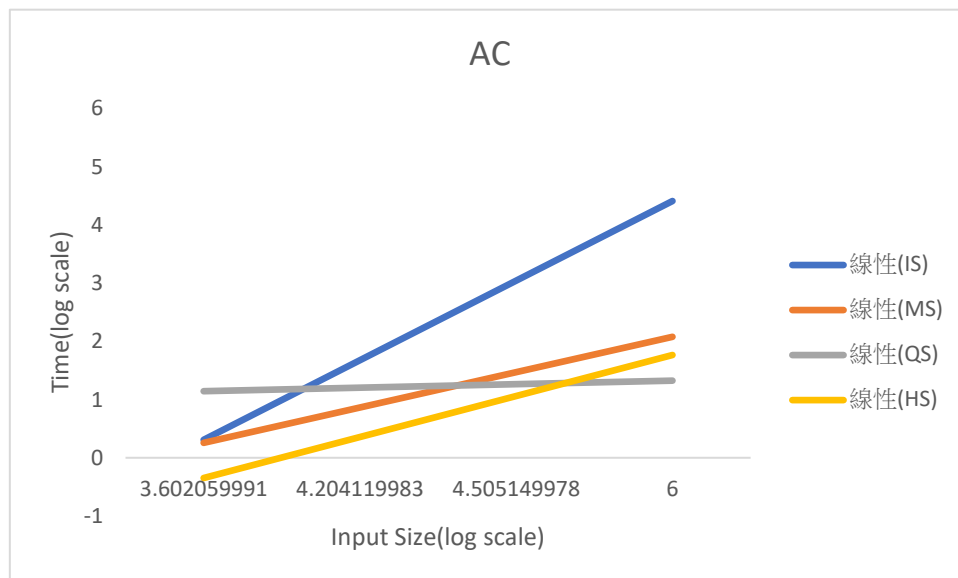


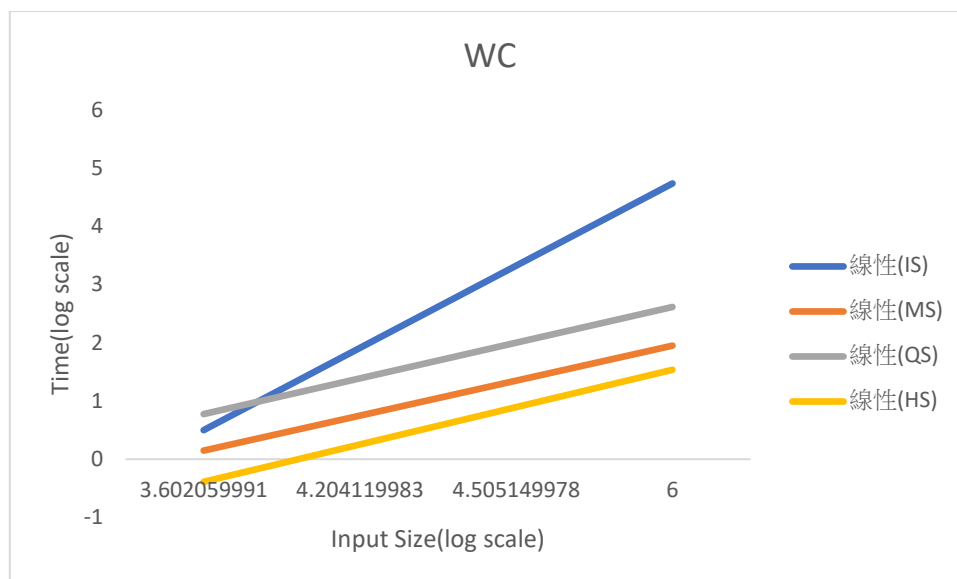
Algorithm PA1

電機四 b08901123 何景盛

Data (running & memory):

Input size	IS		MS		QS		HS	
	CUP time(ms)	Memory (KB)	CUP time(ms)	Memory (KB)	CUP time(ms)	Memory (KB)	CUP time(ms)	Memory (KB)
4000.case2	0.166	5904	2.313	6040	7.645	5904	0.707	5904
4000.case3	7.67	5904	2.306	6040	10.548	5904	0.715	5904
4000.case1	5.215	5904	2.95	6040	10.288	5904	0.816	5904
16000.case2	0.22	6056	4.696	6056	16.687	6056	2.394	6056
16000.case3	63.542	6056	4.511	6056	18.276	6056	1.316	6056
16000.case1	34.761	6056	6.088	6056	19.474	6056	1.667	6056
32000.case2	0.2	6188	6.932	6316	33.894	6188	2.937	6188
32000.case3	243.249	6188	7.813	6316	32.443	6188	2.714	6188
32000.case1	122.413	6188	9.76	6316	30.964	6188	3.56	6188
1000000.case2	1.751	12144	188.451	16236	877.461	12144	78.907	12144
1000000.case3	251544	12144	193.632	16236	965.978	12144	77.363	12144
1000000.case1	124267	12144	266.43	16236	1005.02	12144	141.33	12144





Time complexity:

Algorithm	Worst-case	Average-case
Insertion sort:	$\Theta(n^2)$	$\Theta(n^2)$
Merge sort:	$\Theta(n \log_2 n)$	$\Theta(n \log_2 n)$
Quick sort:	$\Theta(n^2)$	$\Theta(n \log_2 n)$
Heap sort:	$O(n \log_2 n)$	$O(n \log_2 n)$

Analysis:

Average-case:

我們可以觀察到 insertion sort 對比起其他的是最慢的，因為它的 time complexity 是 $\Theta(n^2)$ ，而其他的是 $\Theta(n \log_2 n)$ 。而為什麼 Quick sort 的趨勢對比起 Heap sort 和 Merge sort 有點不同，這是因為我用的是 Randomized Quick sort，所以它的趨勢是大概 $\Theta(n \log_2 n)$ 。

Best-case:

我們可以觀察到 insertion sort 對比起其他的是最快的，因為它在 Best-case 的 time complexity 是 $\Theta(n)$ ，而其他的是 $\Theta(n \log_2 n)$ 。同時我們也可看到 Heap sort、Merge sort 和 Quick sort 的趨勢差不多。

Worst-case:

我們可以觀察到 insertion sort 對比起其他的是最慢的，因為它的 time complexity 是 $\Theta(n^2)$ ，而其他的是 $\Theta(n \log_2 n)$ 。同時我們也可看到 Heap sort、Merge sort 和 Quick sort 的趨勢差不多。原來 Quick sort 在 Worst-case 是 $\Theta(n^2)$ ，但是我現在用的是 Randomized Quick sort，所以在 Partition 中的 pivot 不會每次都抽到最大值或最小值，因此 time complexity 會在 $\Theta(n \log_2 n)$ 。

為什麼我會用 Randomized Quick sort，因為在 worst-case 中，如果用原本的 Quick sort，它的 pivot 會挑最後的 index，所以就會導致 run time 過長，從而令到 core dump。

這些數據都是在 edaU10 上跑的。