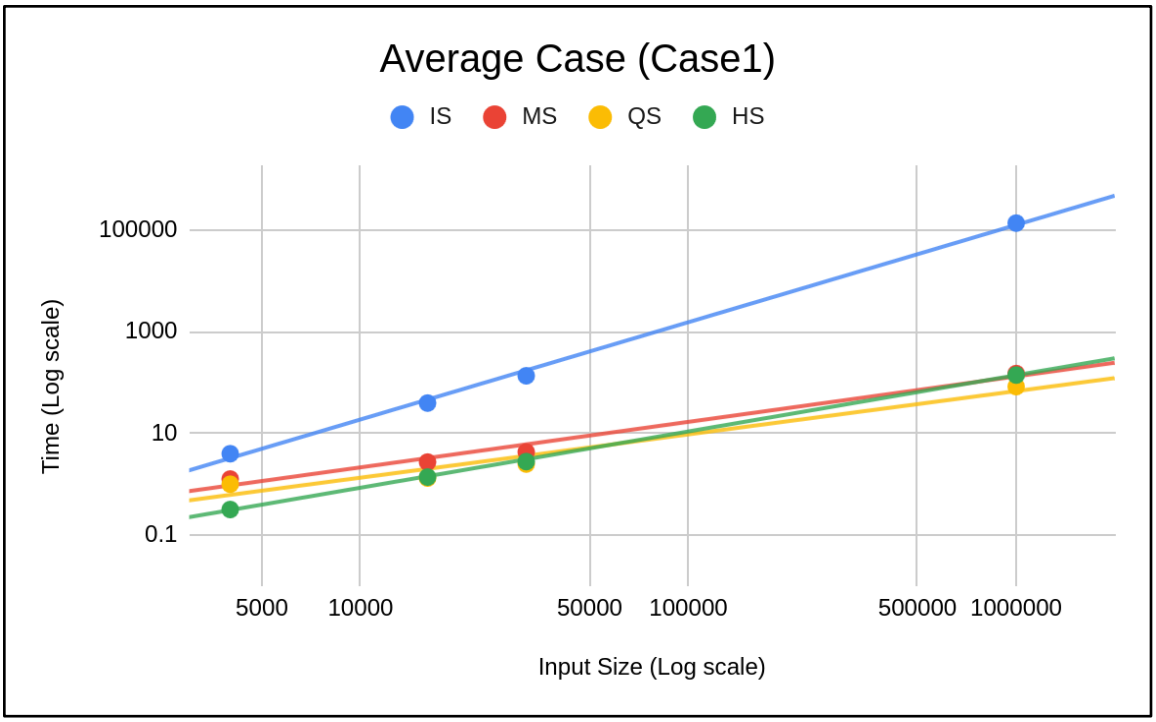
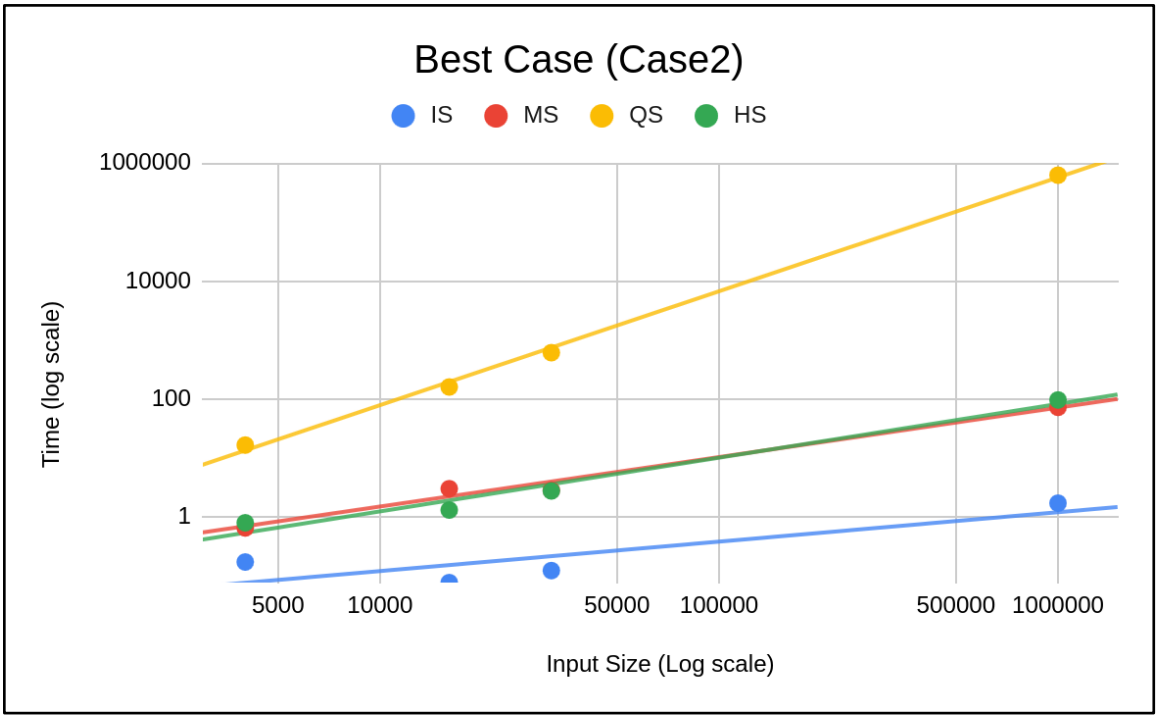
Programming Assignment #1
Sorting

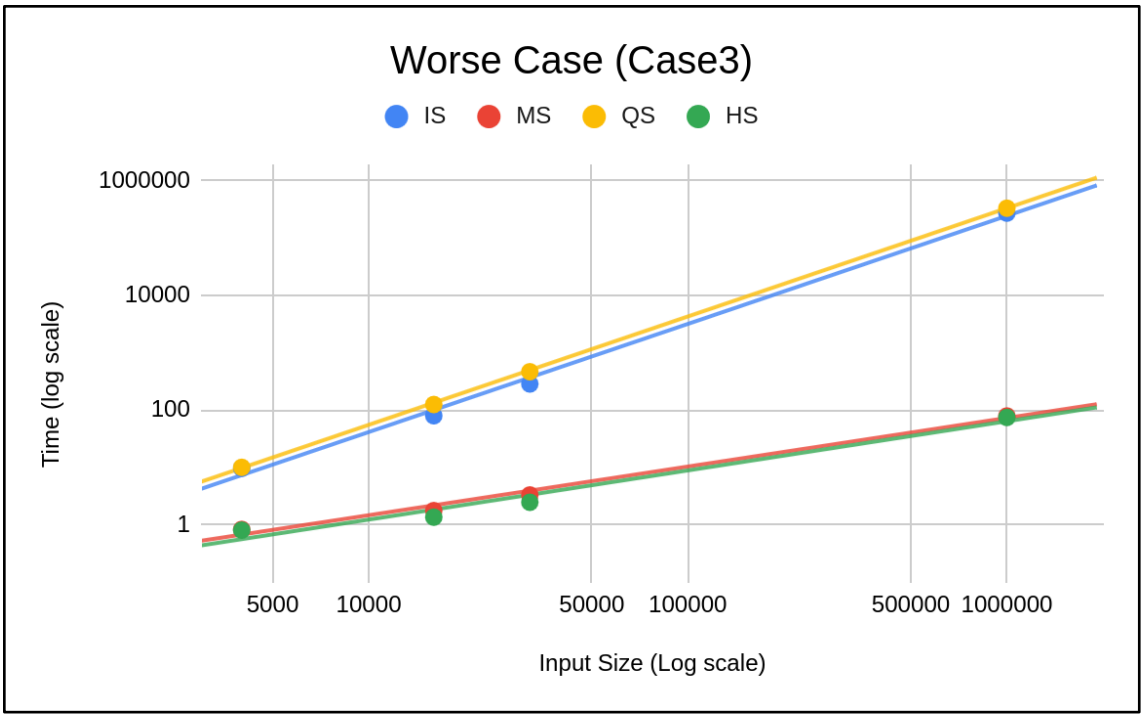
B11901164 陳秉緯

Input size	IS		MS		QS		HS	
	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)
4000.case2	0.172	5904	0.648	5904	16.501	6032	0.792	5904
4000.case3	9.64	5904	0.823	5904	9.964	5908	0.802	5904
4000.case1	3.928	5904	1.237	5904	0.981	5904	0.311	5904
16000.case2	0.077	6056	2.994	6056	158.844	6932	1.306	6056
16000.case3	79.311	6056	1.764	6056	125.007	6432	1.345	6056
16000.case1	39.12	6056	2.681	6056	1.301	6056	1.367	6056
32000.case2	0.123	6188	2.84	6188	608.846	7996	2.761	6188
32000.case3	283.681	6188	3.308	6188	466.604	6984	2.454	6188
32000.case1	134.737	6188	4.254	6188	2.465	6118	2.733	6188
1000000.case2	1.701	12144	72.059	14004	616991	72468	95.781	12144
1000000.case3	273509	12144	78.492	14004	330059	33012	74.056	12144
1000000.case1	138976	12144	149.878	14004	82.61	12144	139.505	12144

Run on EDA union lab machines


Average Case (Case1)
IS
MS
QS
HS
100000
1000
10
0.1
Time (Log scale)
5000
10000
50000
100000
500000
1000000
Input Size (Log scale)


Best Case (Case2)
IS
MS
QS
HS
1000000
10000
100
1
Time (log scale)
5000
10000
50000
100000
500000
1000000
Input Size (Log scale)



IS: The best case is an array that is already sorted(Case 2). Insertion sort has a linear running time (O(n)). During each iteration, the first remaining element of the input is only compared with the right-most element of the sorted subsection of the array so it will be the best one in Case 2. And the other two cases (Case1 and Case 3), the time complexity will be O(n^2). Thus, IS is similar to QS in Case 3 and IS is the worst in Case1.

MS: MS should partition the array into subarray recursively, sort the subarray, and combine finally no matter the case is worst, best, or average. Thus, the time complexity of MS is O(n*lg(n)) in these three cases. And that is the reason why IS (O(n)) will perform better than MS in Case 2.

QS: The worst case occurs when the array is already sorted or reversely sorted, because it will partition into an imbalance tree. The time complexity will be O(n^2). However, when the case is average, the array will be partitioned into a balance tree and it will be O(n*lg(n)). Thus, the run time of QS is both the worst in Case 2 and Case3 and has a similar performance with MS and HS in Case 1.

HS: Need to build a max heap first no matter what the input is and iterate all elements, so the time complexity will be O(n*lg(n)) for Case 1, Case 2, and Case 3. Therefore, the performance is similar to MS in these three cases.