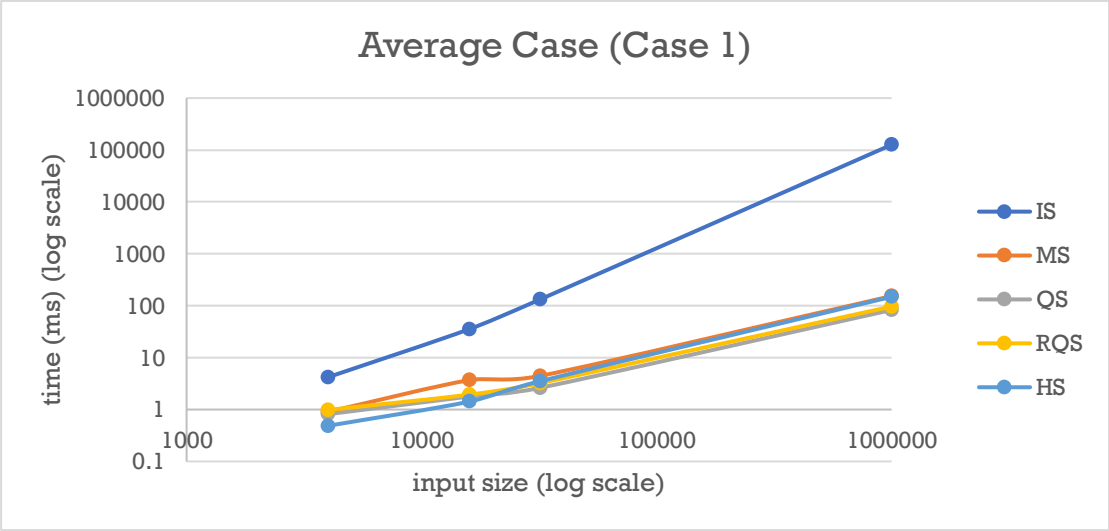
演算法 PA1

B10901041 林新晨

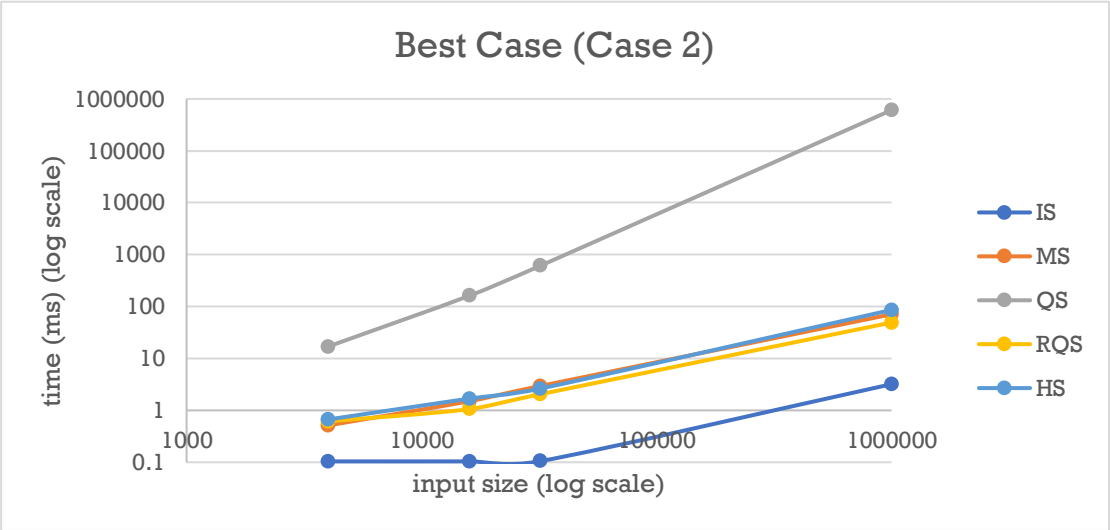
1. Run data on EDA union lab machines.

	IS		MS		QS		RQS		HS	
	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)	CPU time (ms)	Memory (KB)
4000.case2	0.104	5904	0.513	5904	16.927	6028	0.613	5904	0.667	5904
4000.case3	7.732	5904	0.879	5904	12.956	5912	0.645	5904	0.676	5904
4000.case1	4.199	5904	0.906	5904	0.814	5904	0.984	5904	0.486	5904
16000.case2	0.104	6056	1.503	6056	162.135	6932	1.059	6056	1.69	6056
16000.case3	61.271	6056	1.737	6056	122.427	6428	1.056	6056	1.746	6056
16000.case1	35.588	6056	3.693	6056	1.784	6056	1.941	6056	1.425	6056
32000.case2	0.105	6188	2.949	6188	610.608	8000	2.048	6188	2.627	6188
32000.case3	246.489	6188	3.306	6188	460.377	6988	2.388	6188	2.34	6188
32000.case1	132.822	6188	4.451	6188	2.625	6188	3.242	6188	3.523	6188
1000000.case2	3.199	12144	71.032	14004	609602	72472	49.261	12144	86.121	12144
1000000.case3	259703	12144	78.877	14004	341831	33012	51.742	12144	77.364	12144
1000000.case1	126382	12144	153.473	14004	82.889	12144	96.395	12144	148.667	12144

2.



In average case, insertion sort has the worst performance since its average time complexity is $\Theta(n^2)$. Merge sort and quicksort has time complexity $\Theta(n \lg n)$, while heapsort has time complexity $O(n \lg n)$. The time tendency of merge sort, quicksort, and heapsort are the same according to the figure.



In best cases, insertion sort has the best performance with time complexity O(n). Merge sort and heapsort has time complexity O($n \lg n$). Quicksort has the worst performance since its choice of pivot is unwise and has to compare all elements one by one, resulting to longer execution time.

In worst case, quicksort has same time tendency as insertion sort because both algorithms have $\Theta(n^2)$ in time complexity. Both of them are extremely time consuming. Merge sort and heapsort has same tendency since merge sort has time complexity $\Theta(n \lg n)$, while heapsort has time complexity O($n \lg n$).