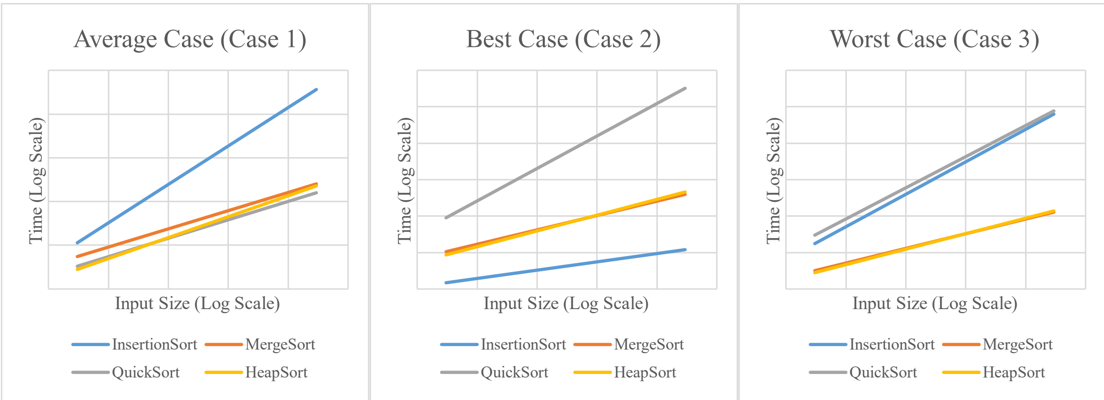
Programming Assignment #1

B11901029 李致頡

Run on EDA Union Lab: edaU12

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	0.125	6072	0.922	6072	18.884	6200	0.768	6072
4000.case3	11.486	6072	0.972	6072	19.484	6072	0.796	6072
4000.case1	6.371	6072	1.775	6072	0.978	6072	0.938	6072
16000.case2	0.176	6224	3.641	6224	134.842	7100	3.328	6224
16000.case3	77.471	6224	3.413	6224	123.938	6596	3.234	6224
16000.case1	44.401	6224	7.222	6224	3.858	6224	3.367	6224
32000.case2	0.200	6356	5.950	6356	488.856	8168	4.641	6356
32000.case3	226.329	6356	5.034	6356	494.398	7156	4.855	6356
32000.case1	137.821	6356	10.474	6356	6.191	6356	5.523	6356
1000000.case2	1.406	12312	77.322	14172	306383	72632	97.864	12312
1000000.case3	180046	12312	86.841	14172	214005	33180	94.297	12312
1000000.case1	89936.1	12312	190.302	14172	108.232	12312	182.841	12312

Trendline plot of run time growth



Theoretical

	Average Case	Best Case	Worst Case
Insertion Sort	$\Theta(n^2)$	$\Omega(n)$	$O(n^2)$
Merge Sort	$\Theta(n \log n)$	$\Omega(n \log n)$	$O(n \log n)$
Quick Sort	$\Theta(n \log n)$	$O(n^2)$	$O(n^2)$
Heap Sort	$\Theta(n \log n)$	$\Omega(n \log n)$	$O(n \log n)$

(Source: https://www.geeksforgeeks.org/analysis-of-different-sorting-techniques/)

Explanation

1. Average Case

   For a random-ordered dataset, Insertion Sort has the longest $\Theta(n^2)$ time complexity, which is consistent with the result of my code.

2. Best Case

   For a sorted dataset, there's no sliding needed for Insertion Sort, so it's the best case and also the fastest ($\Omega(n)$). But for Quick Sort, it has the worst pivot choice, and it's also the slowest ($O(n^2)$) among the four.

   This can be seen from the plot of run time growth above: Quick Sort needs longer time and its trendline slope in log scale is about twice the slope of Insertion Sort. The remaining Merge Sort and Heap Sort both have time complexity $\Omega(n \log n)$, so their slopes are approximately the same and between the two.

3. Worst Case

   For a reverse sorted dataset, it's the worst case for Insertion Sort, since it need to slide every element it had already traversed ($O(n^2)$).
   And for Quick Sort, it again has the worst pivot choice ($O(n^2)$), so Insertion Sort and Quick Sort has about the same slope. Both are larger than that of Merge Sort and Heap Sort.