Algorithm PA1 Report

B06502152, 許書銓

October 14, 2020

---

1 Running Time and Space

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.case1	7.999	12500	3	12500	1.999	12424	1	12500
4000.case2	0	12500	2	12500	1	12424	0	12500
4000.case3	14.998	12500	2	12500	0	12424	1	12500
16000.case1	119.82	12648	8.999	12648	3	12572	2	12648
16000.case2	0	12648	8.999	12648	1	12572	7	12648
16000.case3	241.963	12648	8.999	12648	1	12572	2	12648
32000.case1	481.927	12648	15.997	12836	3	12572	3.999	12648
32000.case2	1	12648	11.999	12836	1	12572	4	12648
32000.case3	786.88	12648	11.998	12836	1.999	12572	4	12648
1000000.case1	283554	18668	442.931	22760	113.983	18592	238.964	18668
1000000.case2	1.999	18668	339.948	27760	33.395	18592	169.974	18668
1000000.case3	567470	18668	333.949	22760	33.395	18592	164.975	18668

2 Running time and Space Analysis

2.1 Case1

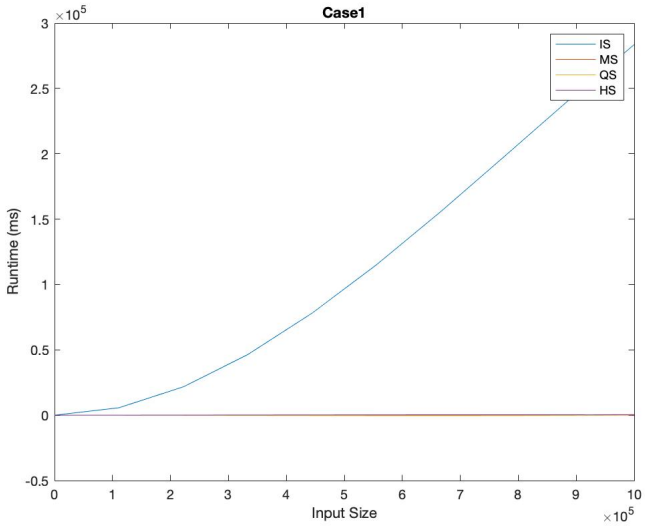

Figure 1: case1

In case1, which is a random array, it is obvious that Inertion sort performs the worst. Because of the $O(n^2)$ algorithm, the runtime will obviously longer than other $O(nlgn)$ algorithms. If we compare $O(nlgn)$ algorithms, the table tells that QS perform the best.

2.2 Case2

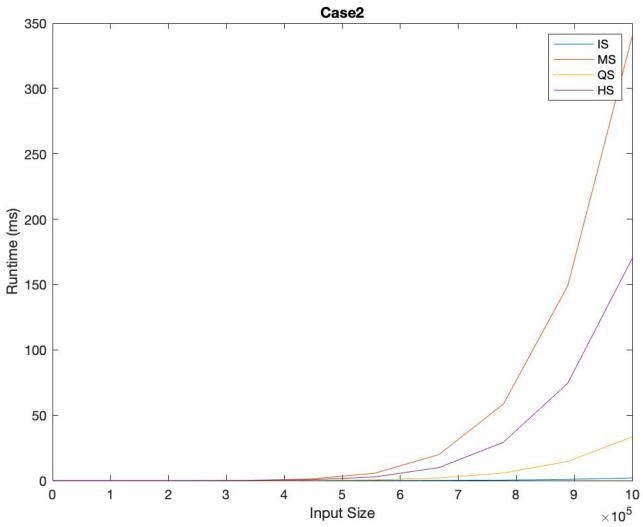

Figure 2: case2

In case2, we can see that algorithms except Insertion sort run the case in same level of time. Moreover, the chart also tells that Insertion sort perform the best owning to this case will be $O(n)$ for insertion sort. Other three methods require $O(nlgn)$ time. Something worth to mention is that i use a median method to choose a pivot for Quick sort; thus, case2 will be $O(nlgn)$ algorithm, which is also best case for Quick sort.

2.3 Case3

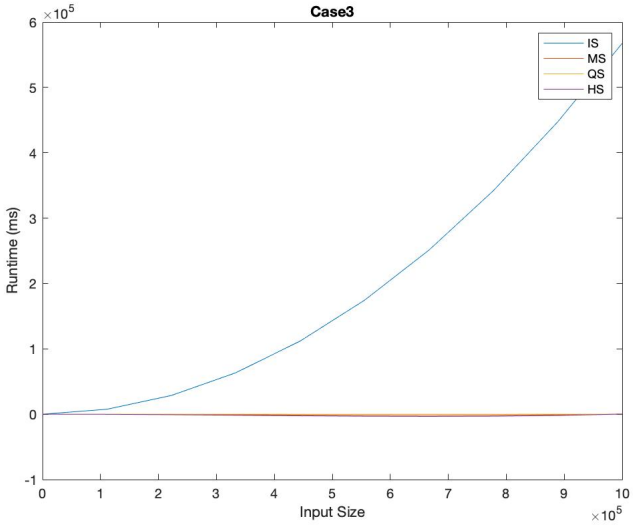

Figure 3: case3

In case3, the chart tells that insertion sort run the worst since its runtime is in the order of 10^5, far longer than other algorithms. Owning to the worst case for Insertion Sort, which is a reversed sorted array, it will take Insertion Sort $O(n^2)$ time. Other three methods require $O(nlgn)$ time. As mention above, I used a median method to choose a pivot for Quick sort; thus, it will be a best case for Quick sort. Hence, Quick sort perform the best.

2.4 Space Analysis

From the upper table, the table tells that Merge Sort use more space than three other method. The Merge process in merge sort need to copy the original array into left and right array; thus, it need $O(n)$ space. While other three method need no extra space to do the sorting process. Hence, other sorting methods require $O(1)$ space.