Algorithms PA1 Report

B08901207 EE3 呂俐君

（A）Data

Using EDA Union port 40051

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	1	12500KB	2	12500KB	0	12500KB	1	12500KB
4000.case3	15.997	12500KB	2	12500KB	1	12500KB	1	12500KB
4000.case1	6.999	12500KB	10.997	12500KB	2	12500KB	1	12500KB
16000.case2	0	12648KB	8.999	12648KB	1	12648KB	2	12648KB
16000.case3	241.963	12648KB	8.999	12648KB	1	12648KB	2	12648KB
16000.case1	120.982	12648KB	10.998	12648KB	0	12648KB	3	12648KB
32000.case2	0	12648KB	18.997	12836KB	2	12648KB	3.999	12648KB
32000.case3	877.866	12648KB	18.998	12836KB	2	12648KB	3.999	12648KB
32000.case1	482.927	12648KB	30.996	12836KB	3	12648KB	4.999	12648KB
1000000. case2	12.998	18668KB	583.912	22760KB	61.991	18668KB	172.974	18668KB
1000000. case3	567304	18668KB	441.933	22760KB	65.99	18668KB	162.975	18668KB
1000000. case1	283767	18668KB	656.9	22760KB	119.982	18668KB	270.959	18668KB

Note:

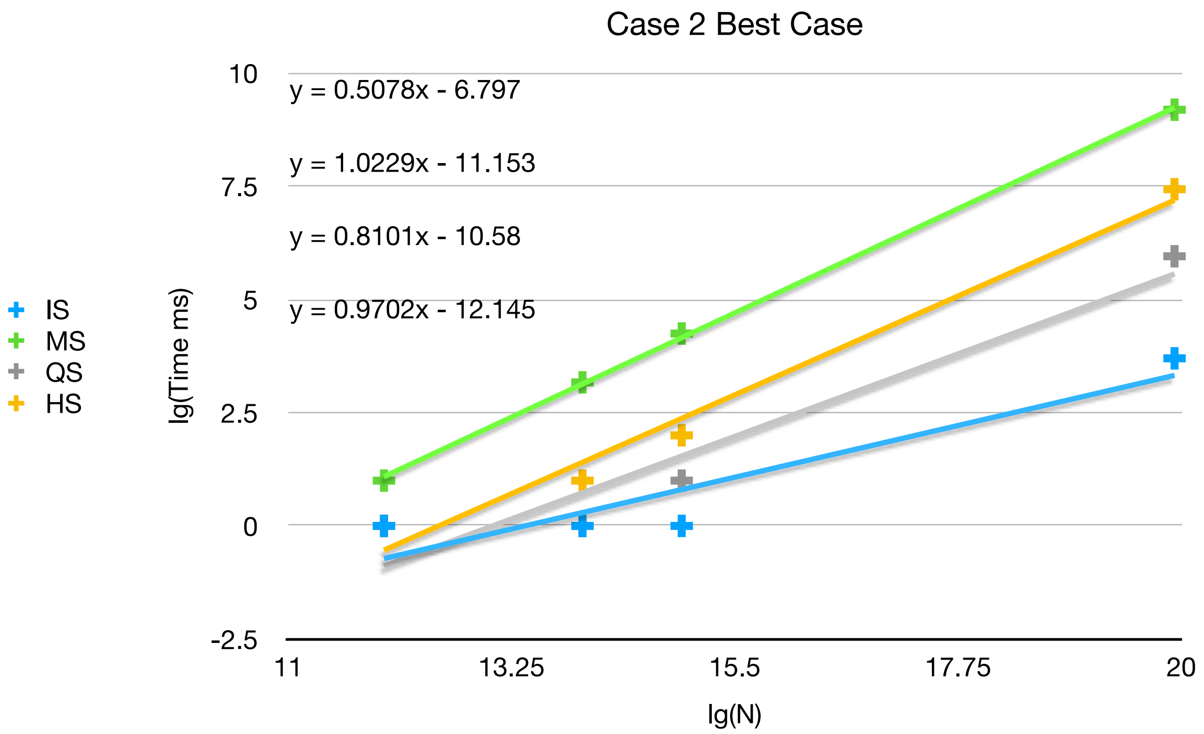
(a) I randomly choose the pivot of Quick Sort in this experiment.
(b) The notation “lg” here means the logarithm base 2.
(c) I set the lg(N) = 0 below when the CPU time = 0ms, since lg(0) does not make sense.

（B）Results

1. Best Case

case 2

N.case	lg(N)	IS (lg(ms))	MS (lg(ms))	QS (lg(ms))	HS (lg(ms))
4000.case2	11.97	0.00	1.00	0.00	0.00
16000.case2	13.97	0.00	3.17	0.00	1.00
32000.case2	14.97	0.00	4.25	1.00	2.00
1000000.case2	19.93	3.70	9.19	5.95	7.43



Discussion：

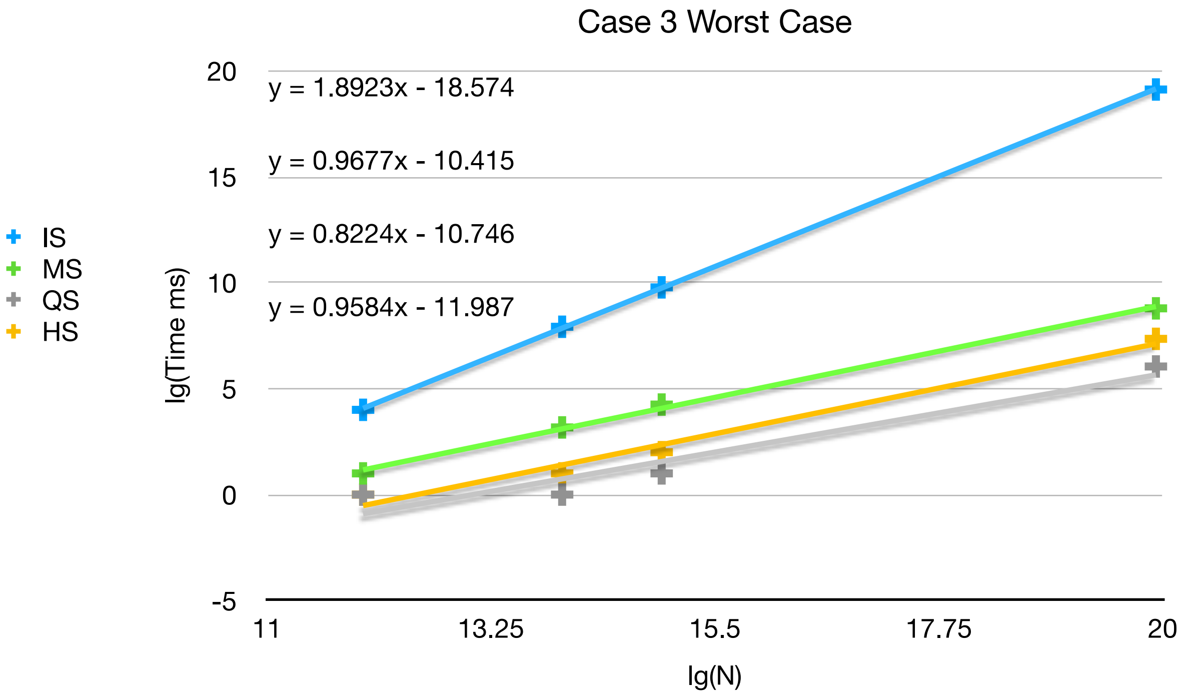
Theoretically,
-IS: Best case time complexity = O(n)
-MS: Best case time complexity = O(n lgn)
-QS: Best case time complexity = O(n lgn)
-HS: Best case time complexity = O(n lgn)

From this graph, we can tell that Insertion Sort under best cases performs most efficiently. Though the rest of three methods seem to be time-wasting under best cases, they are overall quite stable, which we will see in the following experiments.

2. Worst Case

CASE 3

N.case	lg(N)	IS (lg(ms))	MS (lg(ms))	QS (lg(ms))	HS (lg(ms))
4000.case3	11.97	4.00	1.00	0.00	0.00
16000.case3	13.97	7.92	3.17	0.00	1.00
32000.case3	14.97	9.78	4.25	1.00	2.00
1000000.case3	19.93	19.11	8.79	6.04	7.35



Discussion：

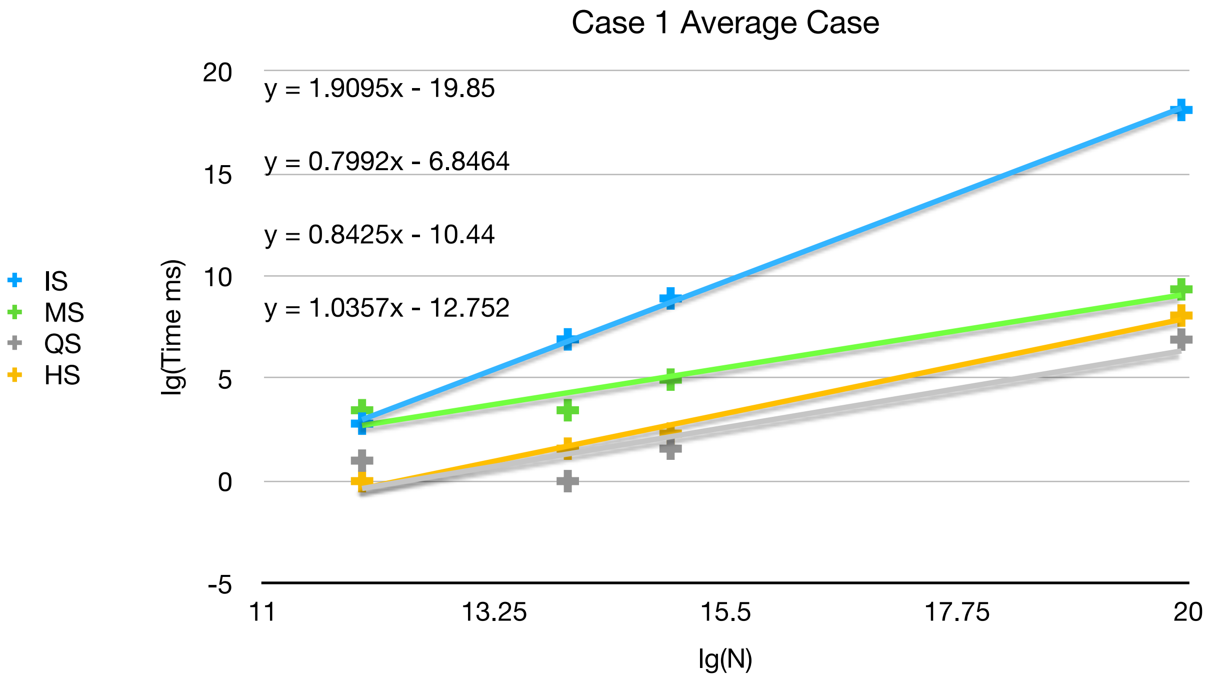
Theoretically,
-IS: Worst case time complexity = O(n^2)
-MS: Worst case time complexity = O(n lgn)
-QS: Worst case time complexity = O(n^2)
-HS: Worst case time complexity = O(n lgn)

In this experiment, Insertion Sort under worst cases performs higher time complexity than other three sorting methods.
The reason Quick Sort does not perform similarly to the theorem is that: I randomized the choice of pivot. Therefore, QS in case 3 and case 1 may have similar results or even better in case 3 by chance.

3. Average Case

Case 1

N.case	lg(N)	IS (lg(ms))	MS (lg(ms))	QS (lg(ms))	HS (lg(ms))
4000.case1	11.97	2.81	3.46	1.00	0.00
16000.case1	13.97	6.92	3.46	0.00	1.58
32000.case1	14.97	8.92	4.95	1.58	2.32
1000000.case1	19.93	18.11	9.36	6.91	8.08



Discussion：

Theoretically,
-IS: Average case time complexity = O(n^2)
-MS: Average case time complexity = O(n lgn)
-QS: Average case time complexity = O(n lgn)
-MS: Worst case time complexity = O(n lgn)

Similar to the case 3 experiment, Insertion Sort has the highest time complexity under average cases while the other three perform more efficiently. We can see that the slope of MS, QS, and HS are only with slightly difference.