

Algorithm PA1 report

李采蓉
心理五 b08207042

October 8, 2023

Processing Data

*running on local terminal

Input Size	IS		MS		QS		RQS		HS	
	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)	CPU time (s)	Memory (KB)
4000.case2	0.000206	5928	0.00115	5928	0.02727	6000	0.000782	5928	0.00083	5928
4000.case3	0.020593	5928	0.001182	5928	0.033037	5928	0.000845	5928	0.000852	5928
4000.case1	0.010293	5928	0.00294	5928	0.001228	5928	0.001268	5928	0.001028	5928
16000.case2	0.000336	6080	0.004389	6080	0.386574	6716	0.003196	6080	0.003789	6080
16000.case3	0.288375	6080	0.003722	6080	0.4196	6336	0.003091	6080	0.003707	6080
16000.case1	0.136622	6080	0.008526	6080	0.003977	6080	0.005051	6080	0.004172	6080
32000.case2	0.000469	6212	0.008621	6212	1.27375	7528	0.00579	6212	0.00764	6212
32000.case3	1.16904	6212	0.008141	6212	1.46883	6764	0.006499	6212	0.007373	6212
32000.case1	0.545764	6212	0.015319	6212	0.008456	6212	0.010078	6212	0.009037	6212
1000000000.case2	0.008047	12168	0.263555	14028	1130.79	56872	0.178921	12168	0.26864	12168
1000000000.case3	739.956	12168	0.260894	14028	722.771	27280	0.186059	12168	0.268388	12168
1000000000.case1	431.03	12168	0.593056	14028	0.321491	12168	0.343535	12168	0.563894	12168

Analysis

Case 1

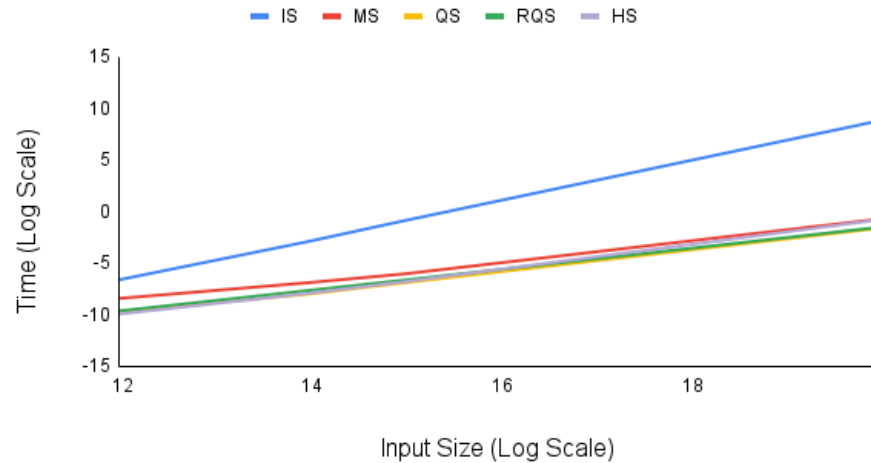


Figure 1: case 1, out of order

Case 1 的測資為亂序，從圖1可見，其中以 Insertion sort (以下用 IS 表示) 最慢，其餘的 Merge sort (以下用 MS 表示)、randomized quick sort (以下用 RQS 表示)、Heap sort (以下用 HS 表示)、Quick sort (以下用 QS 表示) 明顯在測資越大時越相近

在這裡 IS 的速度與 case 3 中相仿，而 QS 的速度是三個 cases 中最快的。

Case 2

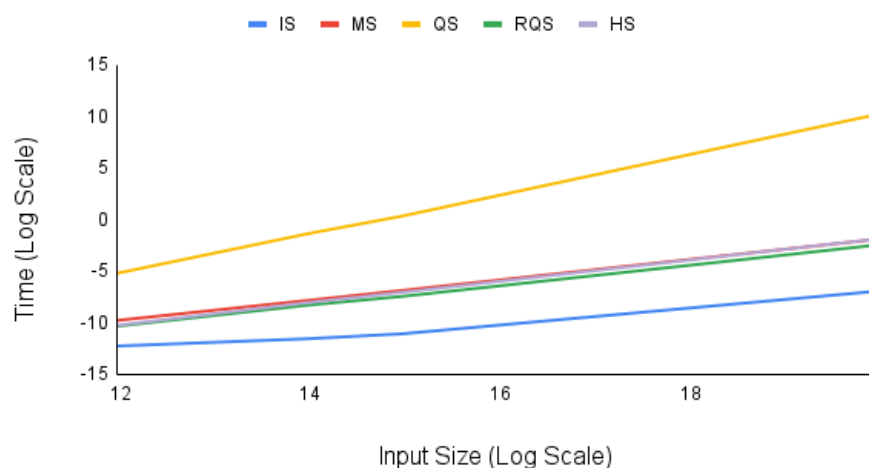


Figure 2: case 2, ascending order

Case 2 的測資皆是由小到大排序，已經排好了所以照理來說不太需要 swap 等的操作，所花時間上理應會是三個 cases 中最少的。從圖2可以看出 IS 的速度最快，MS、RQS、HS 速度差不多，而 QS 速度明顯最慢。

在這裡 IS 的速度是三個 cases 中最快的，QS 則和 case 3 的情況相仿。

Case 3

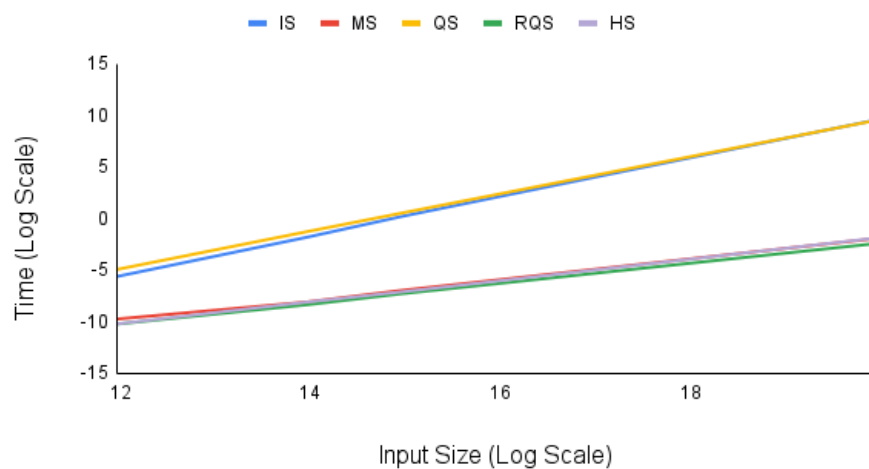


Figure 3: case 3, descending order

Case 3 的測資皆是由大到小排序，在操作時需要把整個序列完全倒過來，預期上會是最花時間的測資。由圖3可以發現速度快慢大致可以分成兩群，一群為較快的 MS, RQS, HS，另群為較慢的 IS 和 QS。

The slope of each curves

對 input size 和 time 皆取 2 為底的對數。

1. 我們可以將所有的 sorting 在不同 cases 下的狀況約略分為三類，時間複雜度分別為 $O(n)$ 、 $O(n \log n)$ 、 $O(n^2)$ 。下面我們對等號兩側取 log，降低縱軸和橫軸各自變項內的差距。

- (a) $t = n \rightarrow \log t = \log n \rightarrow y = x$
 (b) $t = n \log n \rightarrow \log t = \log n \log n = \log n + \log \log n \rightarrow y = x + \log x$
 (c) $t = n^2 \rightarrow \log t = 2 \log n \rightarrow y = 2x$

這是比較理想的情況，現實中時間複雜度往往是 $t = a * n + b, t = a * (n \log n) + b, t = a * n^2 + b * n + c$ 的形式，取 \log 時多少會因為 b 的存在而有誤差。

2. 以下數據根據 google sheet 的函式 SLOPE 所得出，其計算公式為 $s = \frac{\sum_{i=1}^4 (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^4 (x_i - \bar{x})^2}$ ，
 即為最小平方法，而理想的時間由上述的三項公式 (a),(b),(c) 推論

	IS	MS	QS	RQS	HS
Case 1	0.5000000000	0.9167900293	0.9167900293	0.9167900293	0.9167900293
Case 2	1	0.9167900293	0.5000000000	0.9167900293	0.9167900293
Case 3	0.5000000000	0.9167900293	0.5000000000	0.9167900293	0.9167900293

以下則為實際觀測數據所計算出的斜率

	IS	MS	QS	RQS	HS
Case 1	0.5174683286	1.015922375	0.9767502071	0.9840050973	0.8652788861
Case 2	1.402947934	1.014267689	0.5184518814	1.017963394	0.958761445
Case 3	0.5266659857	1.009108922	0.5531259518	1.020333802	0.9609563187

除了 IS 在 case 2 的斜率差異到 0.4 之多外，其餘都大致符合理想狀況的推論。

Difference from the example

- 在 case 3 (圖3) 中，IS 和 QS 有幾近相同的線，這與 example 中不一樣，example 中的 QS 有較小的時間複雜度。由於是倒序的測資，IS 對於當前元素，需要檢查每一個已排序的元素，在歷遍這部分後才能找到適合的插入位置，因此時間複雜度會到達 $O(n^2)$ ；而 QS 則是因為 partition 的過程中皆以最大的元素作為 pivot，導致每次分出的兩個序列都極度不平衡，大大提升時間複雜度，使其同樣需要 $O(n^2)$ 的時間。如果對於 QS 的問題改善狀況的話，可以使用 RQS，讓 pivot 不會必為最大值，規避分割不平衡的風險，從圖3來看的話，RQS 確實能讓時間複雜度降低到 $O(n \log n)$ 。
- 從三個 cases 中可以發現，RQS 無論在何種狀況下所花時間都差不多，約為 $O(n \log n)$ ，至於 QS 則是非常受到排序方式影響，在亂序 (case 1) 的時間複雜度比倒序 (case 3) 跟順序 (case 2) 來的小，圖1可以看到 QS 跟 RQS 幾乎在同一條線上，測資越大越明顯；但在圖2和圖3中，QS 明顯比 RQS 所花時間還要多，這是因為 QS 在這兩種情況下取的 pivot 不是最大值就是最小值，使序列的分割很不平衡，讓時間複雜度上升到 $O(n^2)$ 。
- 在倒序的情況下，當很不幸的 RQS 選到較大的值作為 pivot，所產生的狀況與 QS 相同：partition 不平衡造成時間複雜度上升到 $O(n^2)$ ，這時 RQS 就會跟 IS 有相同的時間複雜度。