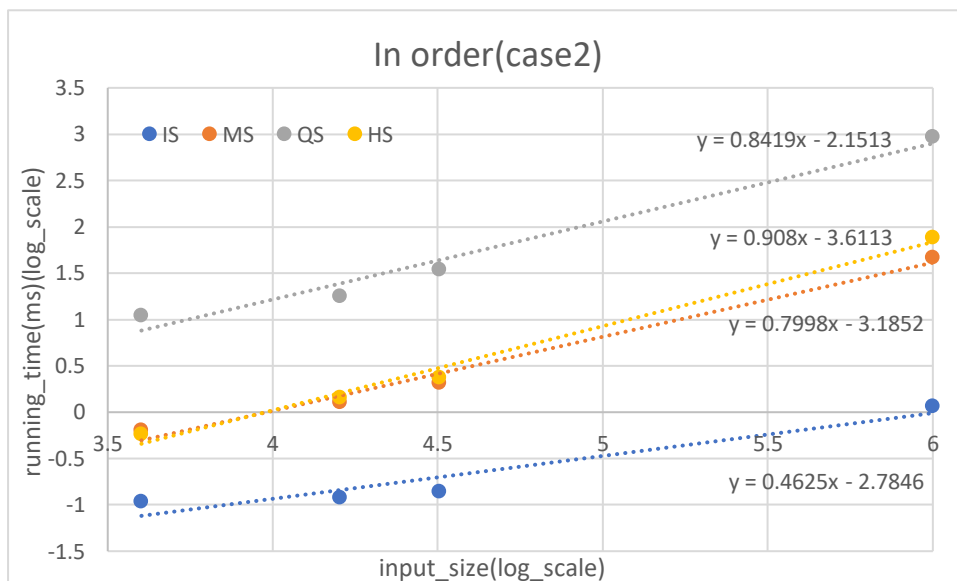
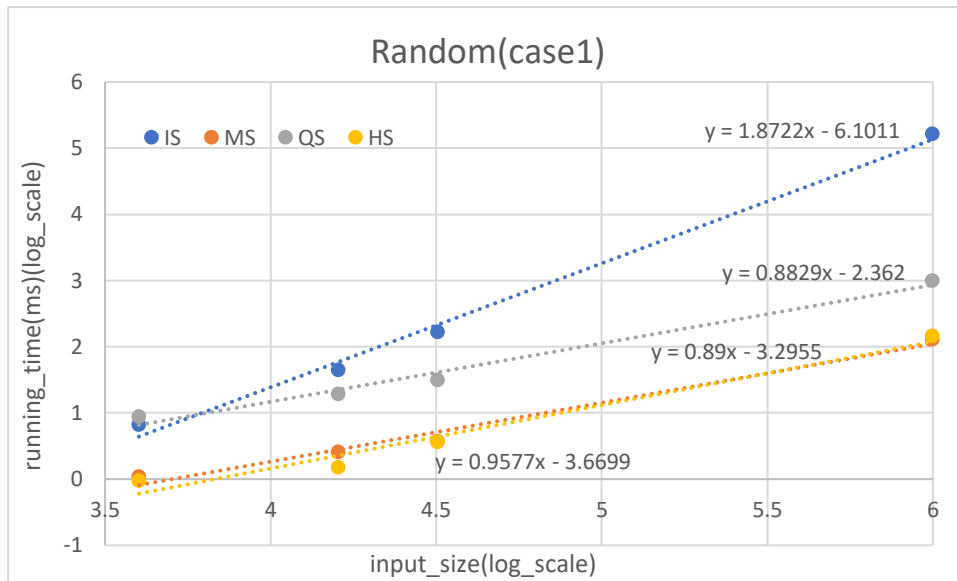
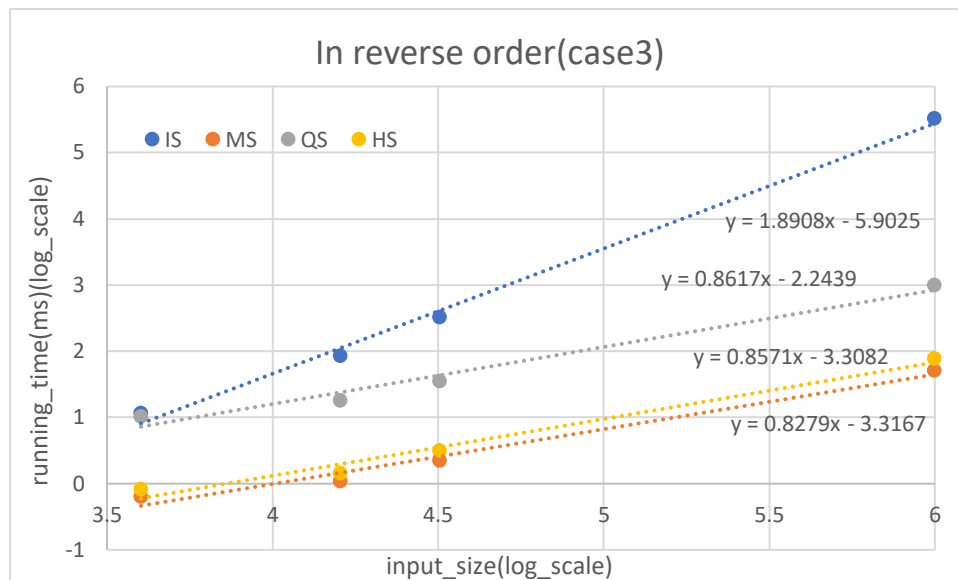


以下數據為程式在 EDA Union port 40056 執行的結果

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	0.109	5904	0.642	5904	11.122	5904	0.576	5904
4000.case3	11.498	5904	0.634	5904	10.293	5904	0.825	5904
4000.case1	6.51	5904	1.068	5904	8.75	5904	0.938	5904
16000.case2	0.12	6056	1.29	6056	17.994	6056	1.437	6056
16000.case3	84.762	6056	1.084	6056	17.902	6056	1.434	6056
16000.case1	43.75	6056	2.571	6056	19.268	6056	1.498	6056
32000.case2	0.14	6188	2.083	6188	34.4	6188	2.37	6188
32000.case3	326.409	6188	2.231	6188	35.068	6188	3.151	6188
32000.case1	163.685	6188	3.697	6188	31.215	6188	3.579	6188
1000000.case2	1.171	12144	46.529	13872	939.446	12144	77.235	12144
1000000.case3	323948	12144	50.962	13872	984.346	12144	77.499	12144
1000000.case1	161956	12144	128.431	13872	992.948	12144	143.286	12144





$$\log(T(n)) = a \times \log(n) + b = \log(10^b \times n^a)$$

所以不同 sorting 方法的趨勢線斜率代表各方法的 running time 大概跟 input size 的幾次方成正比，而 sorting 大多也滿足這樣的關係，所以圖形都非常接近線性。根據三個不同 case 的散佈圖可以看出 Merge sort, Quicksort(Randomized)和 Heap sort 無論在哪種 case 下趨勢線斜率都很接近 1，與理論相符(Merge sort, Heap sort 恆為  $O(n \log n)$ )，Quicksort 在 average case 下為  $O(n \log n)$ ，worst case 下為  $O(n^2)$ ，但是發生機率很低，本次測試沒有出現這樣的趨勢)。至於 Insertion sort，在 case1 和 case3(分別為其 average case 和 worst case)下趨勢線斜率很接近 2，符合理論(average case 和 worst case 下為  $O(n^2)$ )，而在 case2(為其 best case)下趨勢線斜率只有大約 0.5，比理論還要小(best case 下為  $O(n)$ )，我認為是在 input size 不夠大時 Insertion sort 在 best case 下執行的時間太短( $n-1$  次 comparisons)，所以其他部分的程式所花的時間還無法忽略，導致整體的趨勢線斜率低於 1，如果能採用 input size 更大的 case 做測試我相信斜率會越來越接近 1。