Programming Assignment #1

李昀儒 B10901085

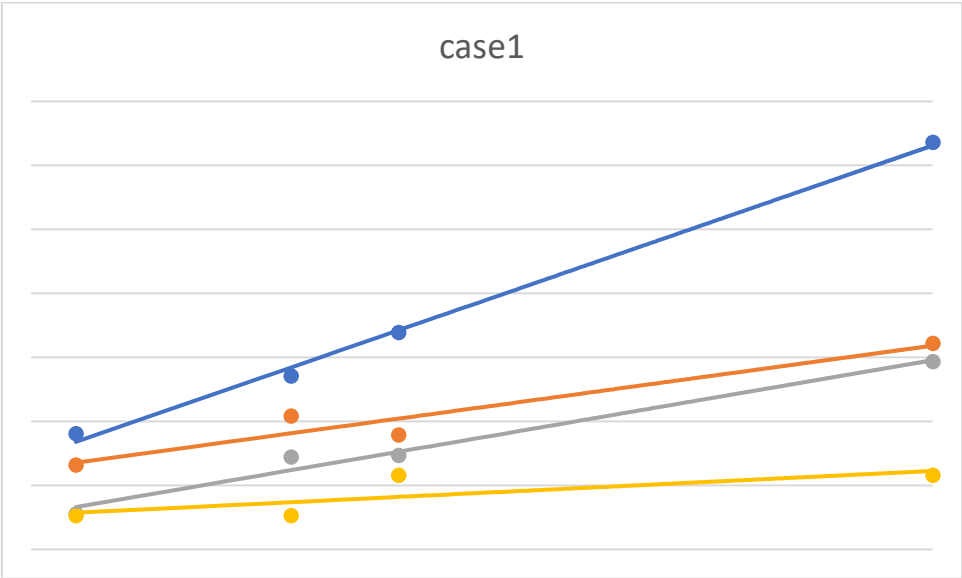
		Insertion Sort		Merge Sort		Quick Sort		Heap Sort	
		CPU time(ms)	Memory(KB)	CPU time(ms)	Memory(KB)	CPU time(ms)	Memory(KB)	CPU time(ms)	Memory(KB)
4000	case2	0.078	5904	0.799	5904	15.712	5972	0.29	5904
4000	case3	8.952	5904	0.618	5904	12.123	5904	0.32	5904
4000	case1	6.368	5904	2.07	5904	0.353	5904	0.335	5904
16000	case2	0.126	6056	1.009	6056	180.412	6680	0.912	6056
16000	case3	132.056	6056	2.354	6056	162.284	6056	1.862	6056
16000	case1	50.877	6056	11.997	6056	2.752	6056	1.423	6056
32000	case2	0.193	6188	4.354	6188	788.597	7500	2.501	6188
32000	case3	532.193	6188	3.019	6188	592.241	6188	3.249	6188
32000	case1	244.176	6188	6.093	6188	2.912	6188	2.895	6188
1000000	case2	1.638	12144	70.352	14004	791689	56844	95.025	12144
1000000	case3	462437	12144	130.11	14004	388472	27256	80.105	12144
1000000	case1	228666	12144	163.817	14004	84.915	12144	161.832	12144

藍:insertion sort

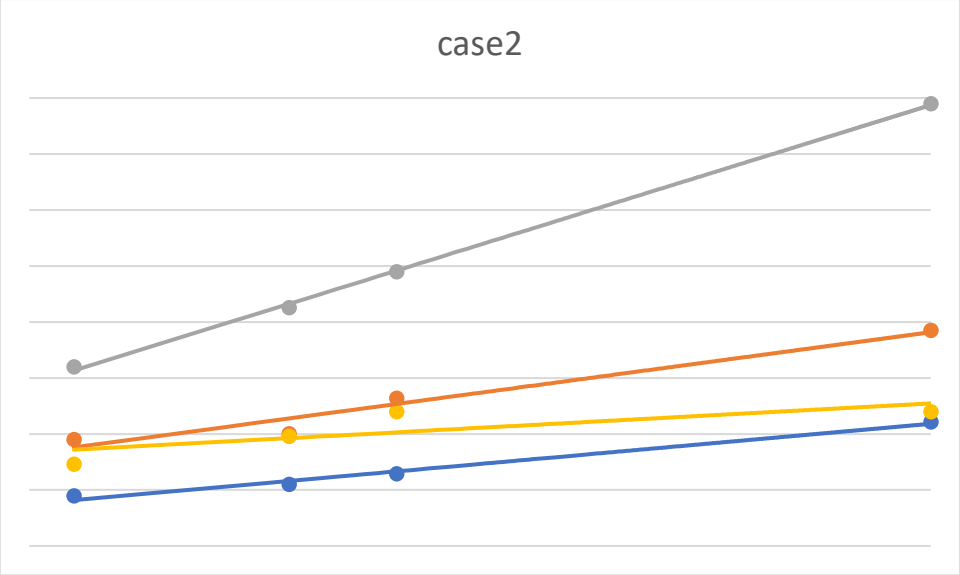
棕:merge sort

灰:quick sort

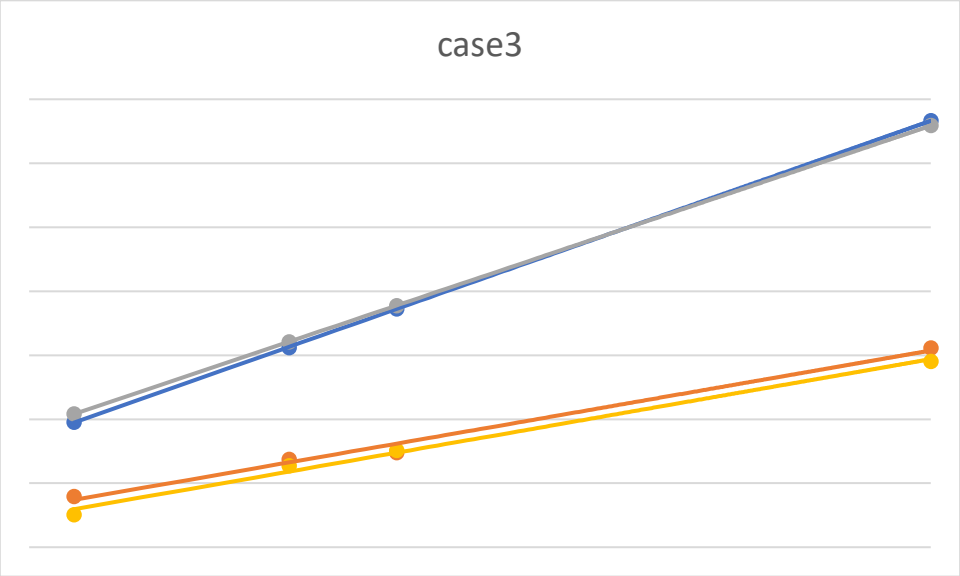
黃:heap sort



Insertion sort 的時間趨勢最大，而其餘三者相近



Quick sort 的時間趨勢最大，而 Insertion sort 的時間尚未和另外兩者拉開



Quick 和 Insertion sort、Merge 和 Heap sort 趨勢類似

1. 這次實作 quick sort 時所選的 pivot 為該 subvector 的最後一個數，所以在 best 和 worst case 時每次呼叫 partition 函數都只會排好一個數，因此在這兩個情況的時間趨勢會和 insertion sort 在 worst case 時相同，可透過隨機選取 pivot number 來解決。
2. 在 case 2 的圖中 insertion 和 merge sort 與 heap sort 的趨勢看起來類似，這是由於資料數量級仍不夠大的關係，但若從表中便可看出 insertion 和其餘兩者已有數十倍的時間差異。