

A、 程式說明:

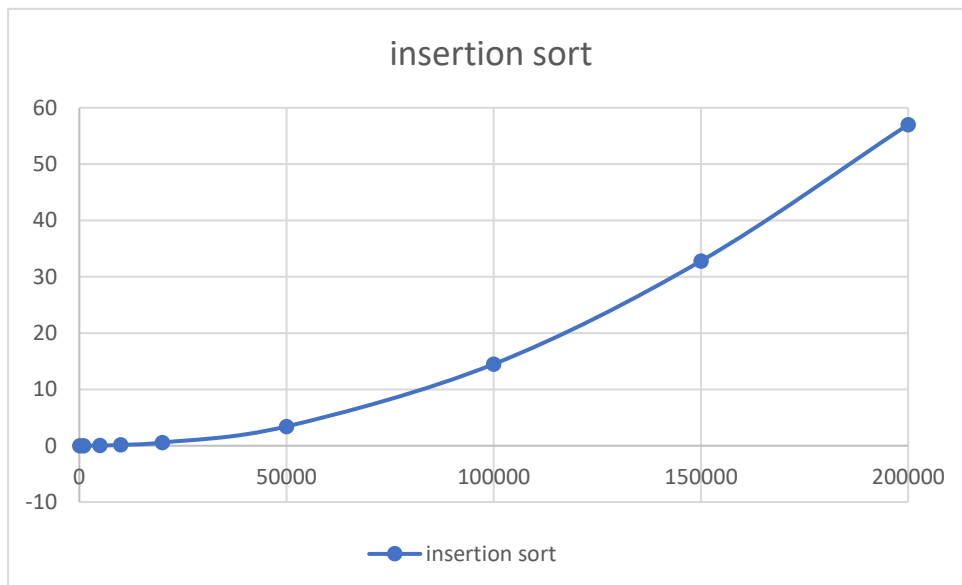
本次作業為比較 merge sort 及 insertion sort，當使用者輸入一個 size(大於 0)時，程式會隨機產生一個大小為 size 的 array，接著為了避免第二次排序時使用已排好的陣列，所以產生一個 array2 複製 array 裡面的資料。最後將兩組相同的資料分別經過 insertion sort 以及 merge sort，分別記錄 insertion sort 和 merge sort 的時間。在執行時，輸入不同 size 分別記錄不同 size 的 insertion sort 和 merge sort 的時間

B、 程式結果:

(單位為秒)

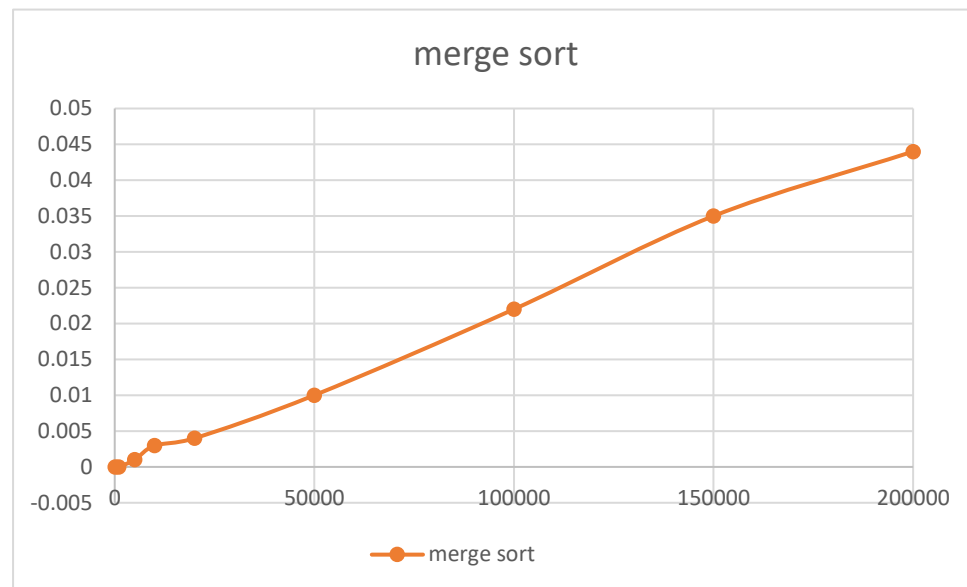
size	100	1000	5000	10000	20000	50000	100000	150000	200000
time(insertion sort)	0	0.002	0.04	0.145	0.562	3.425	14.493	32.793	57.017
time(merge sort)	0	0	0.001	0.003	0.004	0.01	0.022	0.035	0.044

Insertion sort 圖形:



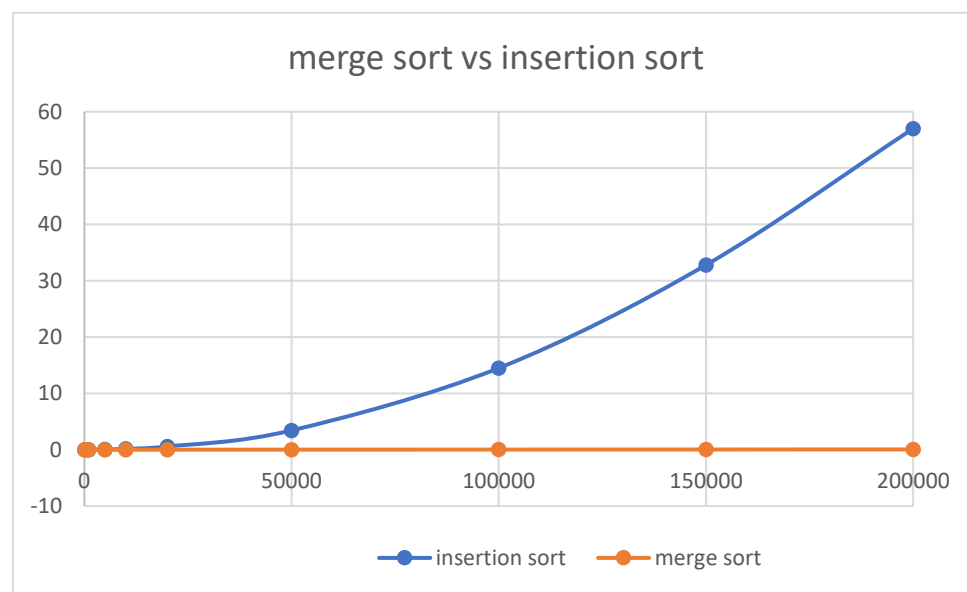
(圖一)

Merge sort 圖形:



(圖二)

Insertion sort 和 merge sort 比較



(圖三)

C、 結果分析:

觀察圖三，我們可以發現 insertion sort 的複雜度明顯大於 merge sort，當 size 變大時，insertion sort 的執行秒數和 merge sort 的執行秒數差異會越來越大。觀察圖一，當 size 變大 10 倍，insertion sort 的執行秒數大約變大 100 倍，所以我們可以得知 insertion sort 的時間複雜度確實為 $O(n^2)$ 。觀察圖二，merge sort 的時間複雜度會小於 $O(n^2)$ 且大於 $O(n)$ ，所以可以得知 merge sort 的時間複雜度確實為 $O(n \log n)$ 。