題目: 分析 sorting 時間

班級: 資訊三丙

學號: D0641915

名字:王千嘉

二、規則說明:

Bubble sort :

1. 比較相鄰的兩個數字，若前面的數字比較大，就互相交換
2. 重複進行步驟(1)直到最大的數字在最後面
3. 重複進行步驟(1)(2)，每次比到上一輪的最後一個
4. 重複進行直到不用比較

Selection sort:

1. 找出為排序中數字最小的數字
2. 將此數字跟已排序的最後一個數字做交換
3. 重複以上直到全部都排序完

Insertion sort:

1. 將資料分成以排序跟未排序的
2. 將未排序中的第一筆，跟已排序的從右到左開始比
3. 直到遇到第一個比自己小的數字，插入到那個位置，並把那個位子後的全部的向右移

Quick sort:

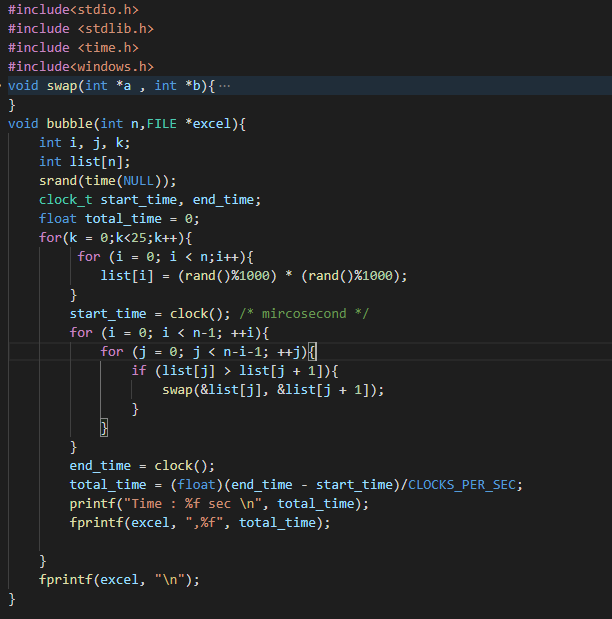
1. 隨便取一個數字，後調整數列
2. 把比這個數字小的都放在左邊，比這個數字大的放在右邊
3. 接著在左邊找一個數字，右邊找一個數字
4. 重複步驟(1)(2)
5. 直到全部的位子的排好為止

Heap sort:

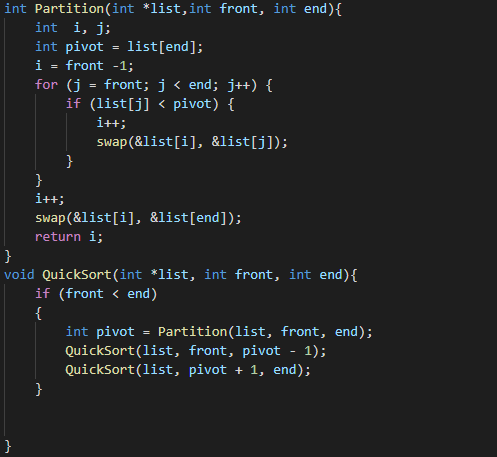
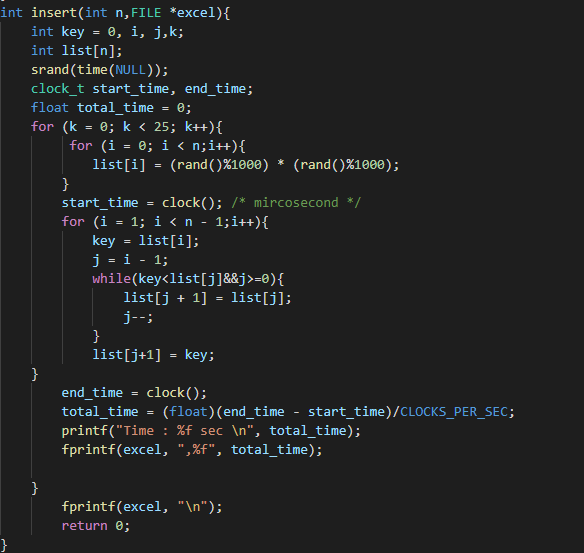
1. 將數列轉換成Max heap
2. 排序 將最後一個節點跟root換，將最後一個節點取出後，並加入已排序的陣列
3. 將整棵樹調整為heap tree
4. 重複步驟(1)(2)

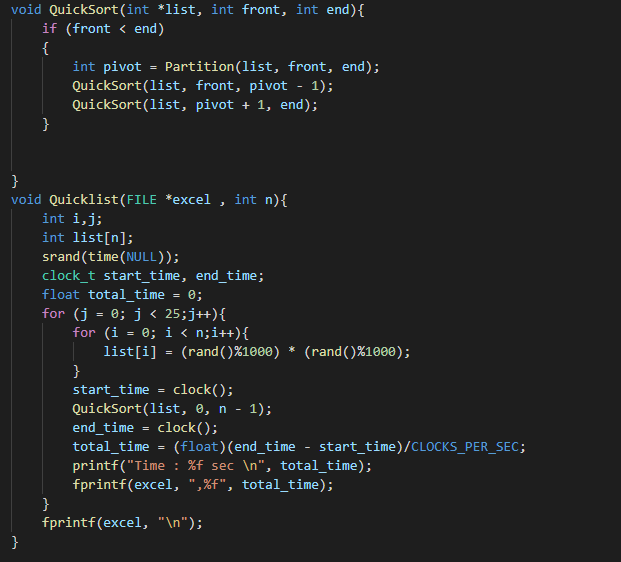
將時間亂數乘上時間亂數降低才重複的機率，並且讓他打破C語言時間亂數只能32000多的限制，將每個要的變數，分別丟進排序裡，用CLOCK()的函式去計算所花的時間，並最後直接輸出成CSV檔。

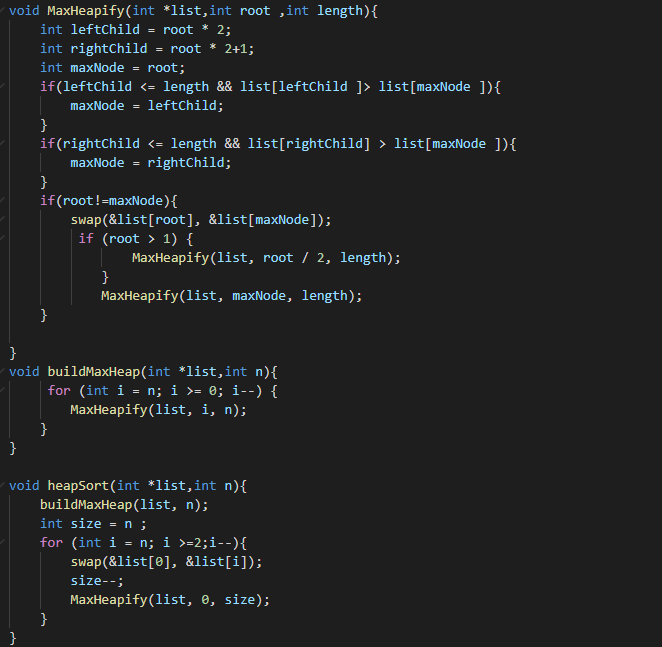
三、程式碼

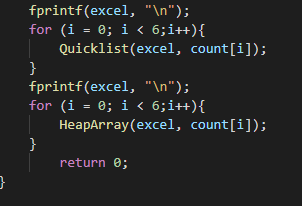
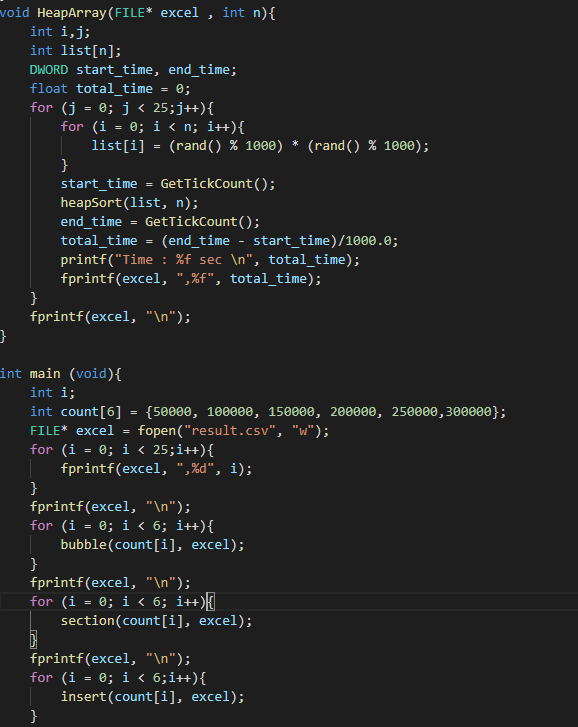












四、執行結果

程式執行過程:

一張含有 建築物, 白色, 電腦 的圖片

自動產生的描述一張含有 電腦, 桌, 白色, 鍵盤 的圖片

自動產生的描述一張含有 電腦 的圖片

自動產生的描述一張含有 黑色, 螢幕, 電腦, 白色 的圖片

自動產生的描述一張含有 電腦, 黑色 的圖片

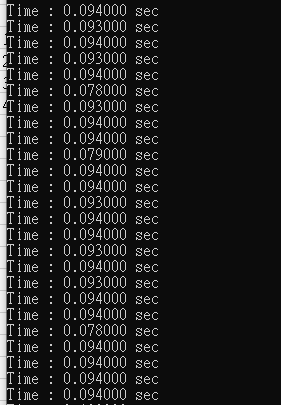
自動產生的描述一張含有 建築物, 電腦, 桌 的圖片

自動產生的描述一張含有 時鐘, 鍵盤 的圖片

自動產生的描述一張含有 建築物, 鍵盤, 時鐘, 電腦 的圖片

自動產生的描述一張含有 鍵盤, 電腦 的圖片

自動產生的描述一張含有 鍵盤, 電腦 的圖片

自動產生的描述一張含有 建築物, 坐, 桌, 白色 的圖片

自動產生的描述一張含有 黑色, 電腦, 桌, 白色 的圖片

自動產生的描述一張含有 桌, 電腦 的圖片

自動產生的描述一張含有 黑色, 電腦, 坐, 黑暗 的圖片

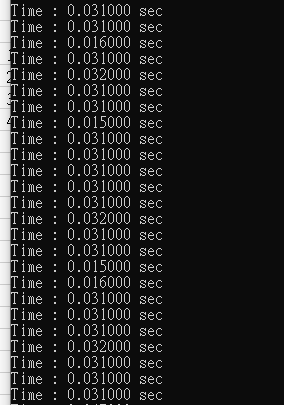
自動產生的描述一張含有 電腦, 白色 的圖片

自動產生的描述一張含有 桌, 電腦, 白色 的圖片

自動產生的描述一張含有 建築物, 超高層建築, 白色, 電腦 的圖片

自動產生的描述一張含有 窗戶, 電腦 的圖片

自動產生的描述一張含有 鍵盤, 電腦 的圖片

自動產生的描述一張含有 電腦 的圖片

自動產生的描述一張含有 鍵盤, 電腦 的圖片

自動產生的描述一張含有 電腦 的圖片

自動產生的描述一張含有 黑色, 電腦, 大, 白色 的圖片

自動產生的描述一張含有 電腦, 黑色, 黑暗, 點燃 的圖片

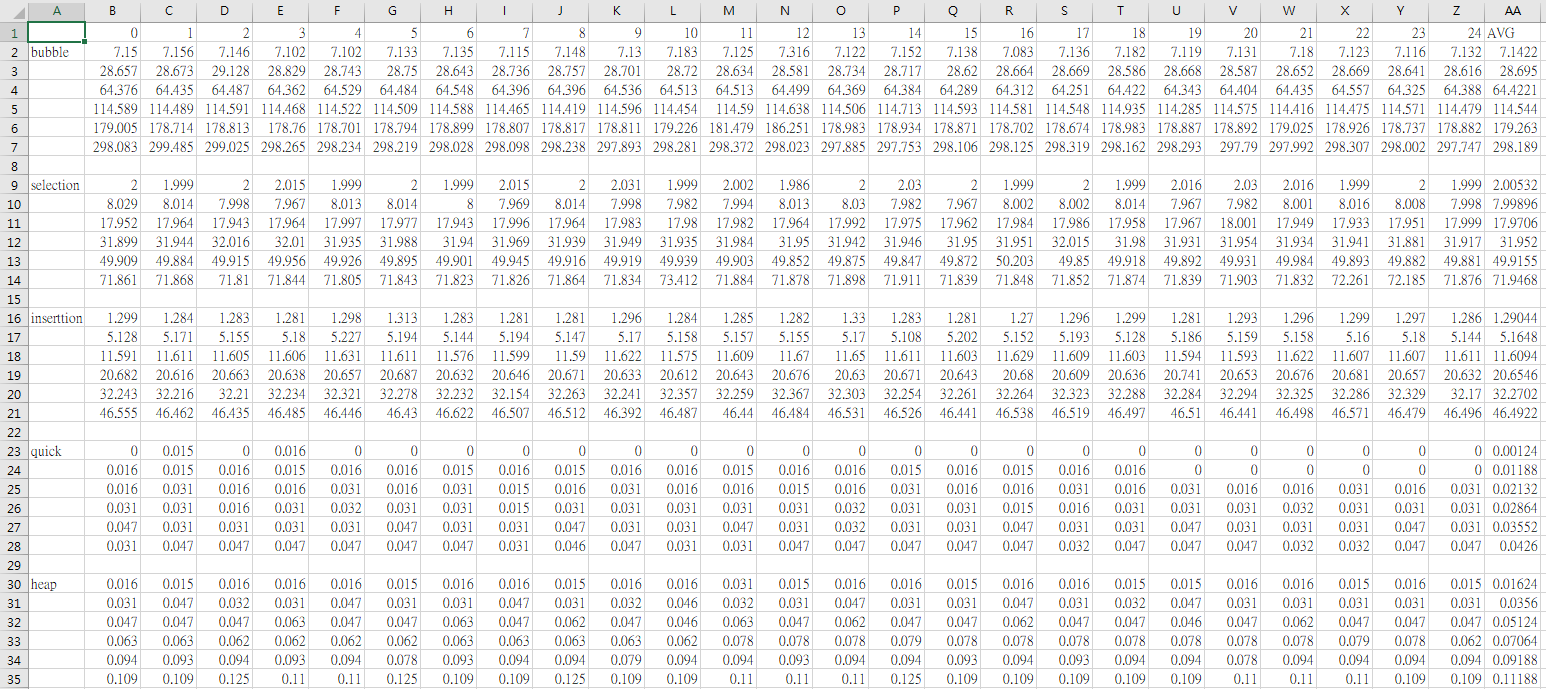
自動產生的描述一張含有 電腦 的圖片

自動產生的描述一張含有 電腦 的圖片

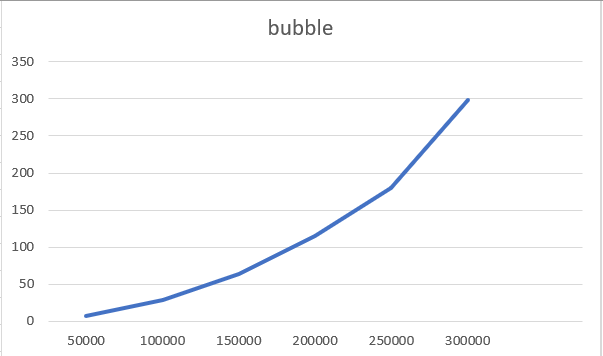
自動產生的描述一張含有 電腦, 白色 的圖片

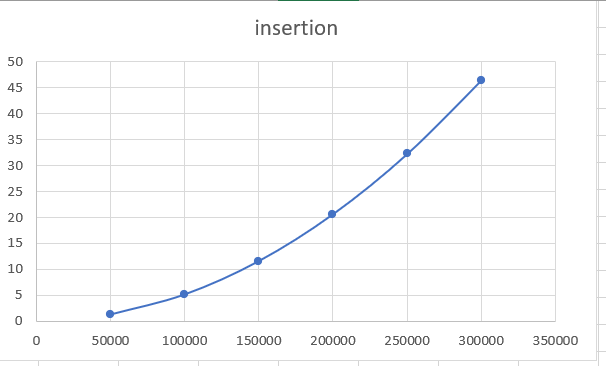
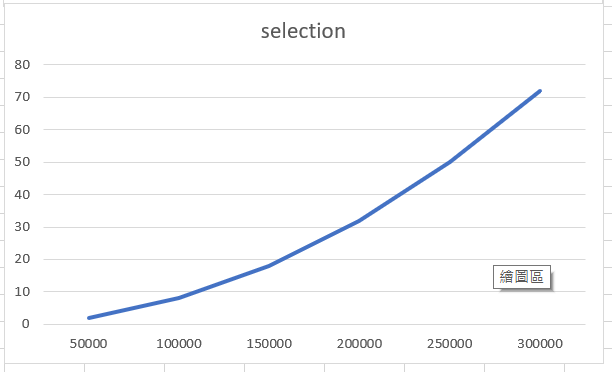
自動產生的描述

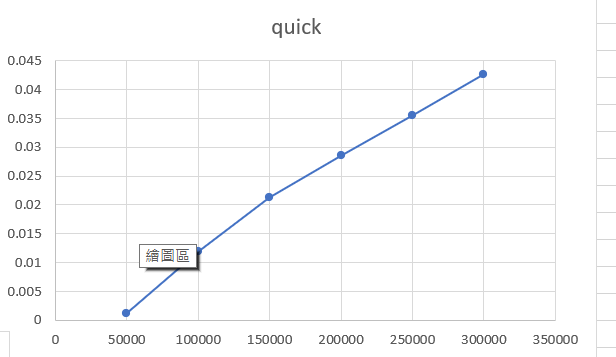
測試數據截圖:(工作表 1)

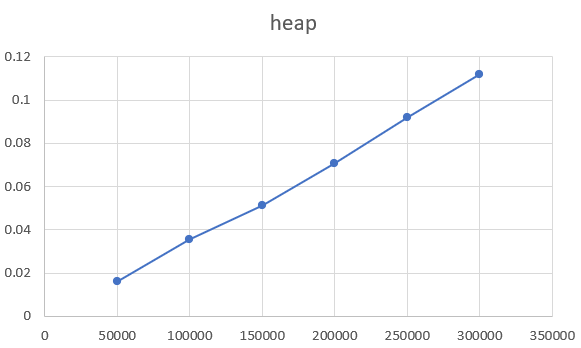


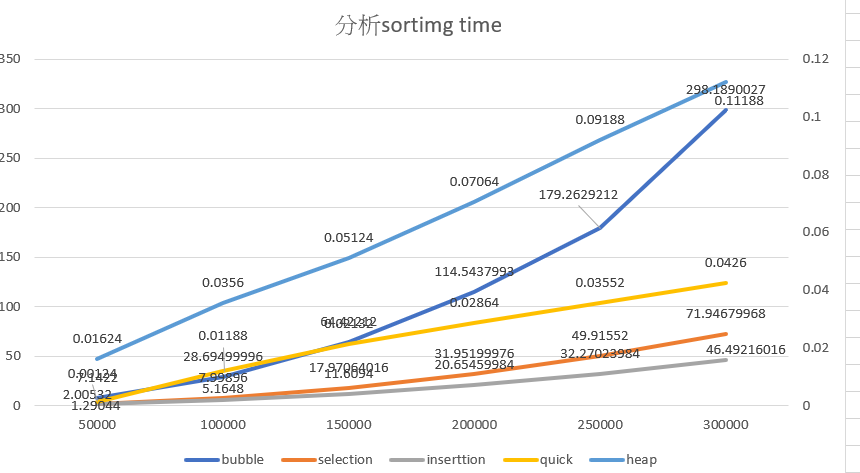
圖表結果:(工作表 2)











五、討論

由實際結果可以排序出所以花的時間，由多到少分別是:

Bubble sort > Selection sort > Insertion sort > Heap sort > Quick sort

Bubble sort 遠遠大於其他四個的排序，並且幾乎有八成的時間都在處理Bubble 的運算。再來雖然Selection sort 跟 Insertion sort 的時間複雜度都是O() 但是Selection 的步驟是 n\*(n+3)/2

，而insertion sort 在最壞情形也是n\*(n+3)/2，所以平均起來它的結果就會小於這個，用數據來證明的結果也跟推出來的理論一樣。再來Quick 跟Heap 兩個都是O(nlogn)可是因為Quick不輸要額外的記憶體空間所以他會稍稍跑得比Heap快一點。

時間複雜度:

Bubble sort() Insertion sort()

Selection sort() Heap sort(nlogn) Quick sort(nlogn)

五、心得:

一直都以為是Heap sort 跑得比 Quick sort 快，但經過這次的實驗後，才發現跟所想的不一樣。再來用C語言寫入時，中間不能直接看結果，要等全部輸出完後才有辦法看結果，這導致再重複輸出了好多次，花費了好多時間，中間再做測試時，還一不小心把數據洗掉，導致前面的時間全部都浪費掉了。第二個遇到最大的問題就是EXCEL輸出結果時，因為Heap 跟 Quick 值太小，導致圖看不出起伏，最後是設雙重Y軸才解決這個問題，主Y軸是以每50個做一個區間，而副Y軸是以每0.02做一個區間，才能有效觀察的趨勢走向。然後畫出折線圖，也是另一個問題，因為對這個東西沒有很熟，所以X軸跟Y軸放相反，圖畫的不好看，不然就是每個折線圖沒有連在一起。