**OS Project1 document**

**資訊三甲**

**10627149**

**許峯僑**

**使用開發環境：**  
**開發軟體:** python jupter // window10

**第一題**，直接將所有資料讀取出來，並且確定資料的筆數後進行bubble sort()排序，並將CPU執行排序時間記錄下來。

**第二題**，先設計一個CutList(fileName, num, f) 將檔案切成k份thread然後分別呼叫bubble sort()，再將k-1個thread進行merge sort() 並且將CPU執行排序時間記錄下來

**第三題**，同上呼叫CutList(fileName, num, f) 將檔案切成k份process然後分別呼叫bubble sort()，再將k-1個process進行merge sort() 並且將CPU執行排序時間記錄下來

**第四題**，同上呼叫CutList(fileName, num, f) 將檔案切成k份然後在一個process內將k份資料呼叫bubble sort()，再將k-1個process進行merge sort() 並且將CPU執行排序時間記錄下來

**分析：**

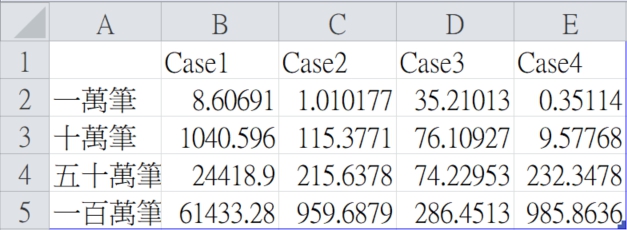
從這次的程式可以真正地了解到演算法分析的Running Time的重要性，相同一個input檔卻會有極大的運算差距。從圖中可以發現最慢的是Case1(bubble sort)因為bubble會將每一筆數據都彼此比較，雖然設計理念十分簡單可以花極短的時間寫出程式，卻會發現當要跑大筆數的數據時會花很多時間。

而case2和case4的運算時間在一百萬筆數據時十分接近，但是在十萬筆的時候有極大的差距一個只要花9秒另一個竟然要花115秒，可能是因為將它分成k個thread做的時間不如直接讓單一process去執行。不過當筆數到五十萬筆、一百萬筆時，兩個的運算時間幾乎一樣。

接著還有一個很有趣的發現，case3分別跑十萬和五十萬時五十萬卻跑得比十萬還快明明是同一隻程式，從中只能分析說兩者的運算時間可能接近，但是在我執行十萬input的數據時電腦可能再做其他事，導致發生這種小bug。

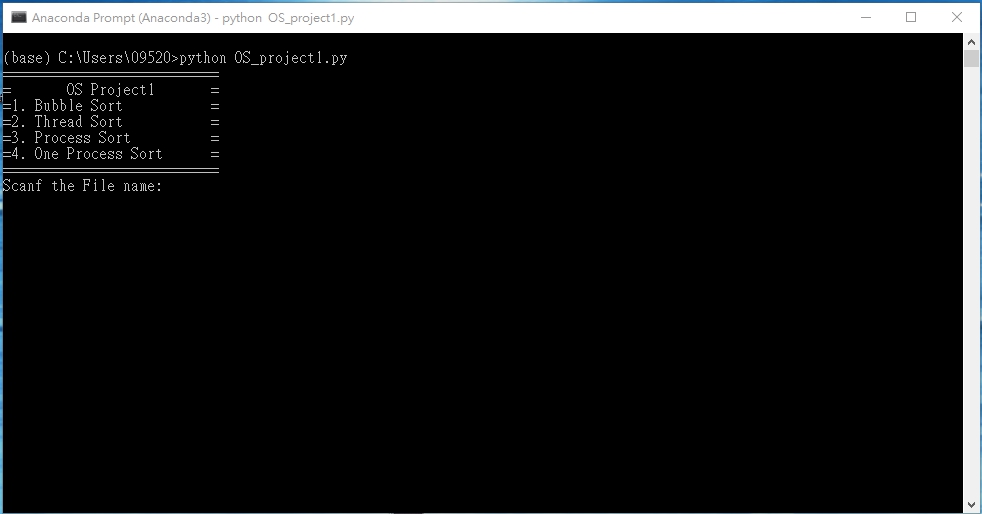
雖然case3在跑一萬筆數據時是四個當中最慢的，但是一百萬筆時卻是四者最快的，從中可以認真去探討甚至判定往後若需要跑大筆數據時可以考慮使用多個process來執行運算。

**Case分別執行不同input的運算時間(秒/單位)**

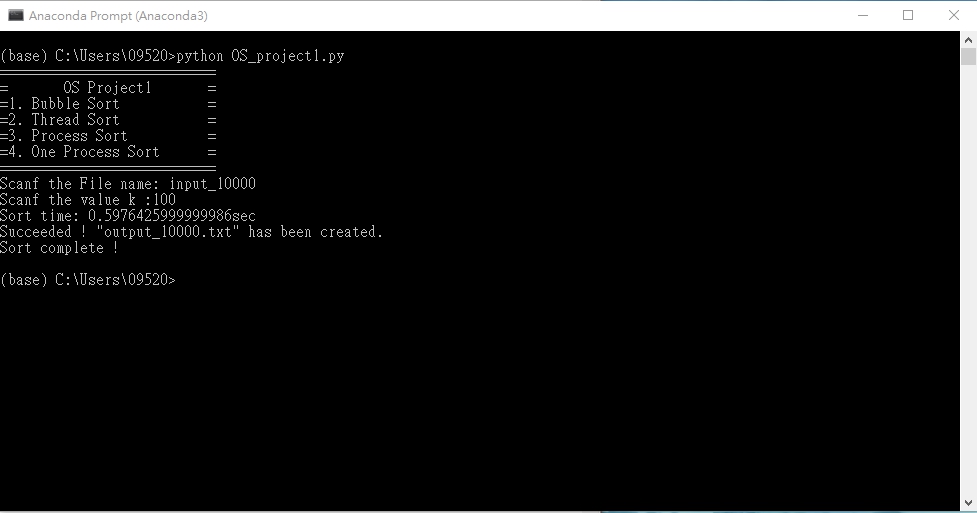


**Case排序運算時間圖表**

**輸入檔案輸入 : input\_xxxxx.txt // xxxx為筆數**



**檔案輸出:** **output\_xxxxx.txt //txt檔的方式輸出結果**



**心得:**

　　在剛開始要撰寫這隻程式的時候一直認為演算法這類學問其實意義不大，想說對於電腦而言跑的速度其實都差不多，不會因為筆數的差距而影響太多最多也頂多2、3秒而已。但是經歷過這次程式才發現原來程式一點點更動其實會大大的影響程式的執行速度。

　　剛開始第一次測試檔案的時候還一度以為我的程式卡在無窮迴圈等了10分鐘怎麼還沒結束也沒跳出結果。一直重新試了很多次最後還是沒辦法處理無限迴圈問題跑去問同學才了解到原來到後期一隻程式要跑的不光光是10幾分鐘，甚至是以幾小時為單位去計算。

　　相信經過這次程式更是讓我們在未來不僅僅是學校的作業，甚至是將來出社會工作時，會謹慎的考慮在甚麼場合有限的條件下去做選擇，究竟是佔有CPU資源重要還是運算的時間重要。