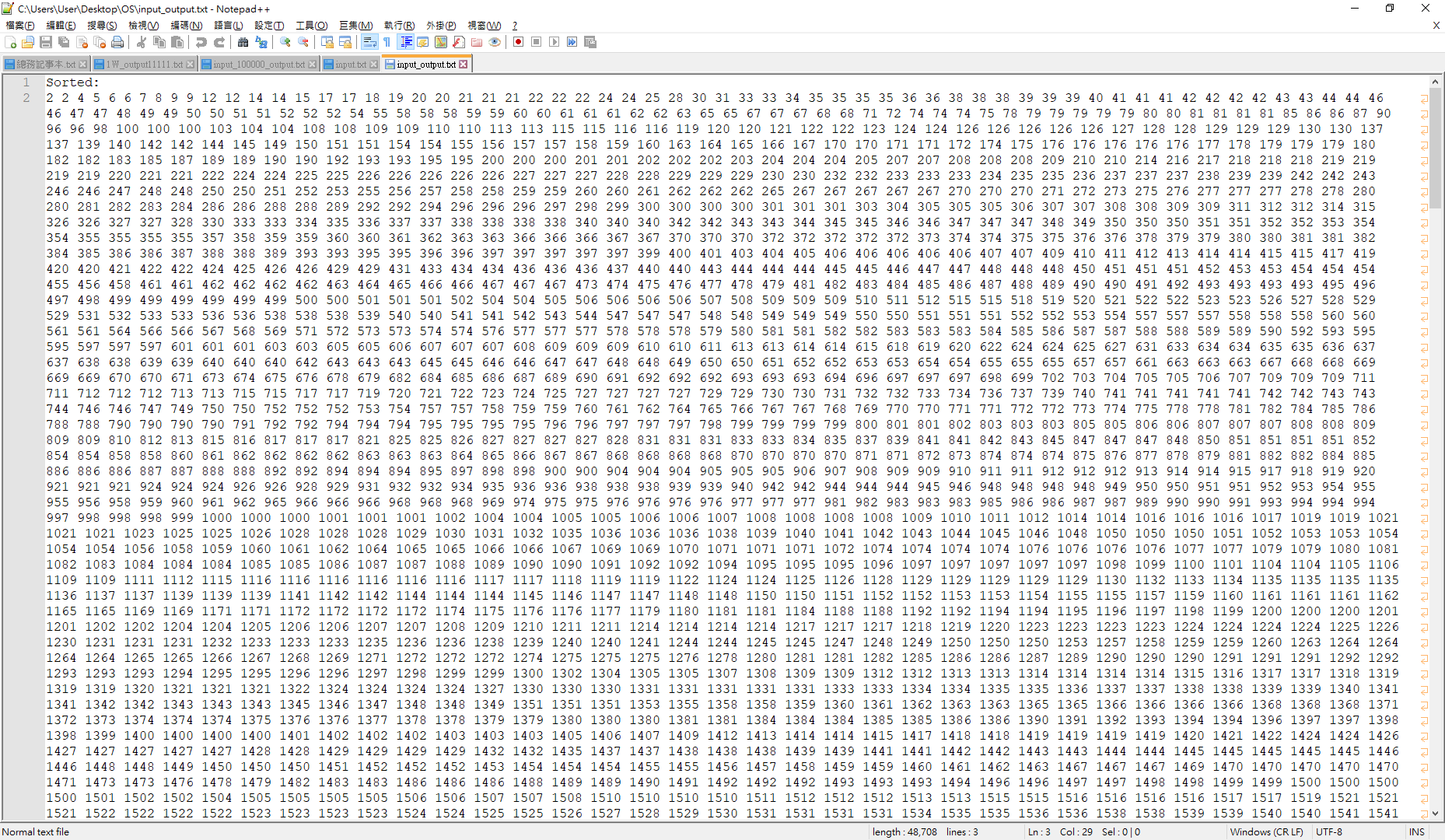
**作業系統第一支程式輸出**

1. **此次程式有四種input，依照第一個數字而不同(投影片內容)，所以程式要自行判別input檔第一個數字為何。執行結果就依照下圖即可****。**

**將排序結果輸出及執行時間。**

input\_output.txt檔 內容範例:



1. **程式說明:**

**使用開發環境：**  
使用軟體: python，一到四題皆完成

程式說明

1.讀資料，切資料(前處理)

2.開始bubblesort

3.依照任務內容做mergesort

4.等CPU跑完

**3.分析：**

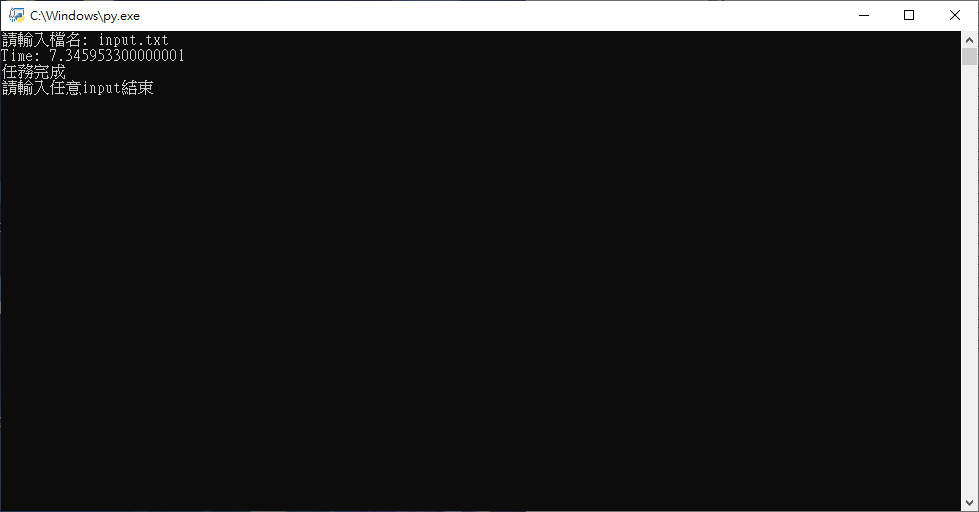
下圖是第一題與第四題分別跑了一萬筆，十萬筆，五十萬筆，一百萬筆的資料，Ｘ軸為資料數，Ｙ軸為執行時間以秒為單位，第一題是單純的只用Bubble sort去跑，第二題將資料使用K個thread分別做bubblesort，第三題則是使用process，第四題是將資料切成K份先各自bubble sort再用一個process merge sort將它處理完畢，由圖表發現，每一種方式資料越多所花費的時間都越多，第一種方式資料越多，時間爬升的越多，但第四種方式，時間爬升的趨於平緩，由此可知第四種方式比第一種好。而第二種與第三種運行模式相似，但使用thread比使用process快上許多，從資料量逐漸上升即可發現差別。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 第一題 | 第二題 | 第三題 | 第四題 |
| 一萬筆 | 7.51784 | 0.17347 | 0.21274 | 0.21235 |
| 十萬筆 | 778.6224 | 7.14615 | 6.43859 | 1.63238 |
| 五十萬筆 | 19465.56 | 40.95303 | 121.0583 | 34.53884 |
| 一百萬筆 | 80613.12 | 70.62303 | 569.3497 | 141.8689 |

輸入檔案輸入 : input.txt

輸出會有: input\_output.txt檔輸出

下圖是第一題範例。



下圖是第三題範例，會再需要輸入integer來使用process

