10527124 邱正皓 資訊三甲

**一、 開發環境:**

作業系統: Windows 10 x64

使用語言: Python

**二、 資料結構 流程**

輸入檔案，依照使用者需求切成k份處理。(有使用jit加速)

Mission1 : 使用read()讀檔並轉成int，然後呼叫bubble\_sort整個丟進去跑。

Mission2 : 使用read()讀檔並轉成int，使用chunkIt分割成k份，以創建k個Thread去做bubble\_sort，再用合併切好的資料兩個兩個去做 k-1 個 thread mergeSort將結果輸出至console並寫出。

Mission3 : import mission3，使用read()讀檔並轉成int，使用chunkIt分割成k份，以創建並執行k個process 去呼叫bubble\_sort，再用合併切好的資料兩個兩個去做 k -1 個 process mergeSort將結果輸出至console並寫出。

Mission4 : 使用read()讀檔並轉成int，使用chunkIt分割成k份，分別去做bubble\_sort，個別排序完後全部合併起來丟進mergeSort再輸出。

相較於M1單純只用一般函式呼叫，M2M3使用multithread與multiprocess，在切4等分的形況下可以將速度推快5~6倍以上的時間但是因為為了避免重疊執行導致錯誤，使用join等待，可能會有點拖慢時間，但依然還是快上許多，而M4相較於M1因為在切分後分別處理bubble單位資料量變小了雖然沒有multi並行下快但相較之下在10000的資料量也快上4倍左右。

**三、圖表分析(單位:秒)**

