資料結構Homework1報告

* 摘要
* 程式規劃
* 實際設計
* 遇到困難
* 參數調整
* 結語

摘要

透過insertion sort(插入排序),selection sort(選擇排序),bubble sort(泡沫排序)及merge sort(合併排序)來對各資料進行排序並分析其花費時間。

程式規劃

一開始的想法就是先把四個排序法寫在最上面，透過亂數生成產生所需的所有參數，並在程式最上方設定資料量(fre)及重複執行次數(smfre)這樣一來再跑資料時即可用更快速的方式來更改參數。在下方我透過switch來分類個排序法，當在終端機輸入1時可產生插入排序法之所有執行後數據，並自動輸入在count.txt裡面以便觀察。

實際設計

一開始我先把需要調整的參數(fre(資料數量) and smfre(執行次數))放在最上面，並開始撰寫三個sort(不含merge)，再來寫出名為out 的function來做資料的測試，最後寫出gogo function來把資料輸出到count.txt(方便使用者閱讀用) and read.txt(繪圖程式用)檔上。Merge sort則是以檔名為001\_merge來做執行為了畫出更精確的圖示我額外寫出一個畫圖用的py檔(draw.py)來協助繪圖。

遇到困難

1. C Language無法放入較大的空間🡪使用new來解決其問題
2. Std::array無法放入較大的空間🡪
3. Merge sort無法再001程式裡執行🡪新增一個程式檔來執行merge sort
4. 0

參數調整

0

0

結語

透過這次的作業我不僅對四種排序法有更深的印象，在C Language array , std::vector and std::array上都有更清楚的了解及應用，並學習到一些平常不見得會使用到的語法。