

# HW3-DP

0616098 黃秉茂

## Environment :

Window10, Code::Blocks 17.12

## Methods or solutions :

利用 ifstream 處理 file 的輸入

利用 ofstream 處理 file 的輸出

透過 getline 取得當行資訊，先將此資訊丟進 string，然後將 string 丟進 stringstream 以取得 data 或轉換型態

Max 函數：回傳較大值

dataset：紀錄 table data 的 vector

dataset\_for\_fisrt\_line：紀錄 table 的第一行 data 的 vector，或是 day，以判斷是否要到 next table

std\_table：行列交換的 table，存放 table data

profitTable：為了解問題的 std\_table 改良版，假設努力天數的回報在努力天數超過最大值時仍為努力天數為最大值的回報

dp：努力多少天所對應的 max score，用在 dynamic programming

沿用 hw1 以 ifstream 和 ofstream 完成輸出入。用 ifstream 和 getline 取得當行資訊，把資訊丟進 string，然後再把 string 透過 stringstream 以取得 data 或轉換型態，接著將所得的資料傳給 dataset。取得 dataset 後，建置一個行列交換的 table(std\_table)。然後要用那一行的 argument 數，來判斷是要透過輸入回答問題(arg = 1)還是要到 next table(arg > 1)，是回答問題的話要無視並跳過下一行(因為是空白行)，並為了解題必須建一個變化版的 table(profitTable：努力天數的回報在努力天數超過最大值時仍為努力天數為最大值的回報)。

這個問題當初卡很久，因為剛開始都用 greedy 的方式去想，發現一些瑕疵之後才發覺是 dynamic programming 中變種的 Knapsack Problem。上網找了很多資料後才知道是類似分組的背包問題，解法大概就是背包問題，再考慮組間的關聯性和衝突。

Reference：<http://www2.lssh.tp.edu.tw/~hlf/class-1/lang-c/DP.pdf>

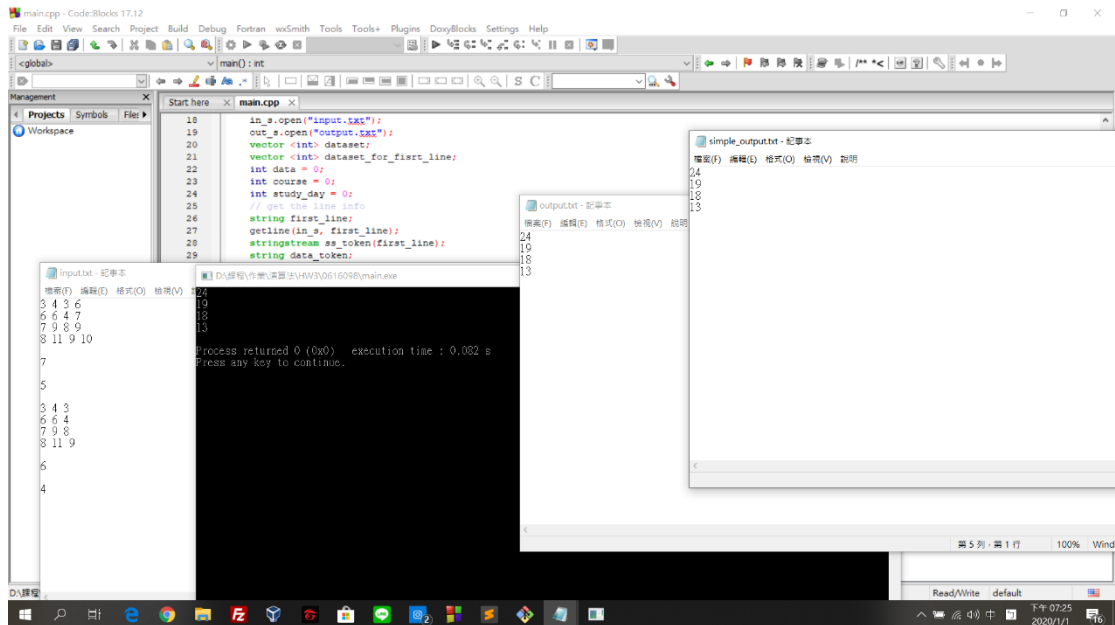
# Results:

input.txt: 範例輸入檔

output.txt: 輸出檔

sample\_output.txt: 範例輸出檔

以 input.txt 為 input 的 output.txt 與 sample\_output.txt 相同 => PASS



用虛擬機的 Ubuntu 的執行結果:

input.txt: 範例輸入檔

output.txt: 輸出檔

sample\_output.txt: 範例輸出檔

以 input.txt 為 input 的 output.txt 與 sample\_output.txt 相同 => PASS

