Data_Structure Hw03 Readme

struct Matrix

```
struct Matrix{
    string name;        //紀錄矩陣名稱
    int N, M, size;     //紀錄矩陣大小
    vector<int> m[3];   //紀錄矩陣裡面的元素
}
```

簡單說明

- 功能1~6都可以使用已設定的矩陣
- 執行程式會有選單告訴使用者每個選項的功能是什麼

set_matrix

- 代入: 矩陣大小(N, M)
- 說明: 設定矩陣(mat)
- 做法:
  - 輸入名稱，會檢查說這個名稱是否有用過了
  - 代入陣列大小(N, M)
  - 輸入元素
  - 回傳 mat
- 複雜度: $O(mat.N * mat.M)$

print_matrix

- 代入: Matrix(mat)
- 說明: 印出所有的元素
- 做法: 矩陣每個元素格子都跑過一次，如果那一個元素有紀錄就輸出，沒有就輸出0
- 複雜度: $O(mat.N * mat.M)$

print_submatrix

- 代入: Matrix(mat), row, col
- 說明: 輸出 Matrix 去掉 row, col 的 submatrix
- 做法:
  - 宣告新的 Matrix(sub)
  - 先判對輸入是否正確
  - 判斷如果都沒有要去掉的話就輸出原本的矩陣
  - 判斷如果有紀錄的那一格元素是在 row 或 col 就跳過
  - 判斷這個元素是在哪個方位左上, 右上(j - 1), 左下(i - 1), 右下(i - 1, j - 1) 然後 push 進 sub
  - 把 sub 代進 print_matrix
- 複雜度: $O(mat.size)$

matrix_transpose

- 代入: Matrix(mat)
- 說明: 轉置矩陣
- 做法:
  - 宣告一個二位陣列
  - 讀取 mat 裡面的元素更新二維陣列
  - 把二維陣列轉成 Matrix 的形式
  - 再把 sub 代進 print_matrix
- 複雜度: $O(mat.N \times mat.M)$

matrix_addition

- 代入: Matrix(A, B)
- 說明: 矩陣加法
- 做法:
  - 輸入兩個矩陣(set_matrix A, B)
  - 判斷兩個矩陣是否符合加法規則
  - 宣告一個新矩陣(add)
  - 矩陣的每個格子都跑過一次
  - 判斷是否有值可以更新
  - 回傳 add
- 複雜度: $O(add.N \times add.M)$

matrix_fastpower

- 代入: Matrix(base), 次方(k)
- 說明: 矩陣冪次方
- 做法:
  - 判斷是否為方陣
  - 初始化答案(ans)
  - 快速冪
- 複雜度: $O((base.N)^3 \times log(k))$

matrix_multiple

- 代入: Matrix(A, B)
- 說明矩陣乘法
- 做法:
  - 判斷是否可以做乘法運算
  - 宣告新的矩陣(ans)
  - 轉置 B 矩陣
  - 先去找是哪個區段的元素要做運算
  - 乘法運算完加總
  - 回傳 add
- 複雜度: $O(A.N \times A.M \times B.M)$