

HW4 README

class Node:

- 紀錄每個節點的row、col、value
 - 每個節點有Node * 型態的變數 Next、Down、Right指向別個Node

class Mat:

- 儲存每個Node
- 讀入矩陣**NodePointer mread(int rows, int cols, int num):**
 - 在mread中輸入矩陣
 - 參數是矩陣的行列個數，以及總共有幾非零元素
 - 然後啟動mread_initialize去做矩陣初始化。在這個過程做了以下幾件事：
 1. 設定FIRST NODE
 2. 替header node new一塊Node型態的記憶體，大小為Max{rows, cols}
 3. 為current_col, current_row new一塊Node型態的記憶體，大小為Max{rows, cols}
 4. current_col每個元素會指向col中的最後一個Node, current_row同理。
 5. 初始化時，current_col與current_row每個元素都指向header node。ex: current_col[i] = &headernode[i] (current_row同理)
 6. 將FIRSTNODE 與所有headernode串起來。
 - 初始化後就讓使用者輸入資料，並使用mread_addNode建立新的Node，建立新的Node的過程如下：
 1. current_row[newnode->Row()]->Right = newnode 將新的Node所應屬於的row最後一個Node的right指向新的Node。
 2. current_row[newnode->Row()] = newnode 將指向「新的Node所應屬於的row最後一個元素」的指標，移動到新node上，使current_row[newnode->Row()]繼續指向最後一個Node。
 3. current_col 同理。
 4. 在輸入的過程只考慮由上而下，由左而右。否則就需用insert into linked list的方式。
 - 所有的非零元素都讀取完後，就會呼叫mread_final將各個row和col最後一個Node的link接回headernode。
- 輸出矩陣**void mwrite()**
 - current_hd指向FIRSTNODE->Right，當這個headernode底下的所有Row都拜訪過後current_hd就會指向當前headernode->Next。迴圈停止於current_hd == FIRSTNODE
 - O 指向current_hd->Down，將當前Node的資料輸出後就會指向當前Node的Down。迴圈停止於O == current_hd。
- 轉置矩陣 **Mat Transpose()**
 - 先設定新矩陣的FIRSTNODE。
 - 利用mwrite()尋訪每個Node的方式逐一利用新矩陣的mread_addNode加入新的矩陣
 - 最後用新矩陣mread_final的方式將新矩陣的row與col關閉。

- 矩陣相乘 **Mat operator * (Mat mat2)**

- 當原矩陣的Col != Row時會raise exception。
- NodePointer hdptr1會指向原FIRSTNODE->Right，NodePointer hdptr2 則指向mat.FIRSTNODE->Right。
- 原矩陣一個Row會與mat2所有Col做相乘，相乘方式：
 1. mat1_element_ptr, mat2_element_ptr分別指向hdptr1->Down,與hdptr2->Right。
 2. 當mat1_element_ptr->Col() == mat2_element_ptr->Row()代表這兩個元素可以相乘，相乘後紀錄結果。
 3. 當mat1_element_ptr->Col() > mat2_element_ptr->Row() 則mat1_element_ptr向下移。
 4. 當mat1_element_ptr->Col() < mat2_element_ptr->Row() 則mat2_element_ptr向右移。
 5. mat2一個Col所有的Node都尋訪完後，如果Sum不為0則new Node，將新的Node存入stack。hdptr2位移到下一個Col，直到所有Col都尋訪完後迴圈結束。
- 當mat1一個Row尋訪完後，hdptr1則位移到hdptr1->Next。
- 當所有相乘的過程都結束後，建立新的Mat物件，從stack——把Node取出，使用mread_initialize與mread_addNode與mread_final把stack中的Node填入新的Mat中。
- 最後回傳新的Mat。