

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

LAB9

有向無環圖與拓撲順序

長榮大學
資訊工程學系

班級：資工 2B

姓名：郭智榮

學號：109B30612

日期：2022/05/16

版本校定紀錄：

版本	更新紀錄	發佈日期
0.0.0.0	初版完成	2022/05/16

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

一. 實驗需求：

(一). 題目說明

描述：

實作完成對任意 DAG 找出拓撲順序之演算法。

輸入說明：

第一行是兩個正整數 n 與 m ，代表節點數與有向邊數，節點是以 0 到 $n-1$ 編號。

接著有 m 行，每一行兩個整數代表有向邊連接的兩個節點編號。

輸出說明：

每筆測試資料輸出一行 DAG 之拓撲順序。

(二). 演算法

i. 虛擬碼

定義 using namespace 型態的 std；//無此定義則 vector 宣告將報錯

```
int main(){
```

宣告四個整數(int)型態變數 n , m , now , end ，分別用於儲存測資節點數量、有向邊數量、佇列(路徑)當前檢查位置及目前最後位置；

輸出提示訊息「請輸入節點數及有向邊數量」；

輸入兩整數代表該筆測資節點數及有向邊數，並存入 n 和 m 中；

宣告一個 vector 型態變數 $vertex$ 並指定存放整數(int)型態的資料，大小為 n ，用於存放該節點與其可前往的節點資料；

建立兩個整數陣列 $haveEntrance$ 及 $path$ ，分別用於記錄可通往該節點的節點數量及拜訪的路徑，兩陣列大小皆為 n ；

```
for i = 0 ~ i < m (for i++){
```

宣告兩個整數型態變數 a 和 b ，用於紀錄本次輸入的有向圖；

輸出提示訊息「請輸入第 i 條有向邊」；

輸入兩整數，並存入 a 和 b ，代表 a 可以前往 b ；

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

```

        將 vertex[a] 存入 b；
        將 haveEntrance[b] 加 1；
    }

    for i = 0 ~ i < n (for i++){
        如果 haveEntrance[i] 為 0{
            // 代表沒有節點可以進入節點 i，因此節點 i 必須為起點
            如果 end 為 0 代表目前路徑是空的{
                將 path[end] 設為 i；
                將 end 加 1；
            }
            否則{
                輸出訊息「該資料沒有路徑」；
                // 有兩個節點必定為起點，因此無法形成路徑
                return 0;
            }
        }
    }
}

while(now 小於 end){
    宣告整數型態變數 v，並設為 path[now] 代表此次檢查的節點；
    將 now 加 1；

    for int u : vertex[v]{
        // 將 vertex[v] 內的值依序放入 u 中

        將 haveEntrance[u] 減 1；
        如果 haveEntrance[u] 為 0{
            // 代表若此次不進行，則後續將沒有節點可以造訪 u
            將 path[end] 設為 u；
            將 end 加 1；
        }
    }
}
}

```

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

```

    如果 end 小於 n 代表路徑內有缺少節點{
        輸出「該資料沒有路徑」；
    }
    否則{
        輸出「路徑為」；
        for i = 0 ~ i < n (for i++){
            輸出 path[i]；
        }
    }

return 0;
}

```

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

二. 完整程式碼：

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int n, m, now = 0, end = 0;
    // 宣告整數變數，分別用於儲存「測資節點數量、有向邊數量、佇列當前檢
    // 測的位置及佇列的終點」
    printf("\n Input number of vertex and edges : ");
    // 輸出提示訊息
    scanf("%d%d", &n, &m);
    // 輸入節點數量並存入 n ； 輸入有向邊數量並存入 m

    vector<int> vertex[n];
    // 建立一個 vector 型態陣列，內部儲存值為整數型態
    int haveEntrance[n] = {0};
    // 建立一個整數陣列，紀錄其他點可以進入該點的數量，預設為 0
    int path[n];
    // 建立整數陣列儲存當前路徑(佇列)

    for (int i = 0 ; i < m ; i++) {
        int a, b;
        // 建立兩整數，用於儲存該有向邊的起點與終點
        printf("\n Input the edge-%d : ", i);
        // 輸出提示訊息
        scanf("%d%d", &a, &b);
        // 輸入有向邊起點並存入 a ； 輸入有向邊終點並存入 b
        vertex[a].push_back(b);
        // 紀錄 a 節點可以通往 b 節點
        haveEntrance[b]++;
        // 將可進入 b 節點的有向邊數量 + 1
    }

    for (int i = 0 ; i < n ; i++){
        // 有 n 個節點，因此 for 迴圈執行 n 次
```

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

```

    if (haveEntrance[i] == 0){
        // 如果可進入該節點的有向邊數為 0，代表該點只能做為起點
        if(end == 0){
            // 若 end 為 0 代表目前路徑內沒有其他節點
            path[end] = i;
            // 在當前路徑的尾部存入 i
            end++;
            // 路徑的尾部位置 + 1
        }
        else{
            // 若 end 不為 0 則代表已有其他也必須做為起點的節點
            printf("\n The data no path.");
            // 因有至少兩個節點都需做為起點，因此本測資沒有答案
            return 0;
        }
    }
}

while (now < end) {
    // 當佇列的當前位置比終點小，代表還有節點需要檢查
    int v = path[now];
    // 將節點設為當前位置的節點
    now++;
    // 當前位置 + 1，代表位置往後 1 格

    for (int u : vertex[v]) {
        // 將 vertex[v] 內值依序帶入變數 u
        haveEntrance[u]--;
        // 節點 u 的入口數量 - 1
        if (haveEntrance[u] == 0){
            // 若節點 u 的入口數量為 0 代表此次不進入，之後就無法進入
            path[end] = u;
            // 在當前路徑最後的位置新增節點 u
            end++;
            // 路徑最後的位置 + 1
        }
    }
}

```

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

```


    }
}

if (end < n){
// 當路徑最後的位置比節點總數小時進入 if
    printf("\n The data no path.");
    // 因為最後位置比節點總數小，代表有節點未被新增，因此不符合要求
}
else {
    printf("\n Path : ");
    // 輸出路徑訊息
    for (int i = 0 ; i < n ; i++){
        // 因有 n 個路徑，因此 for 迴圈執行 n 次
        printf("%d ", path[i]);
        // 輸出路徑節點
    }
}
return 0;
}

```

資訊工程學系	資料結構應用	文件編號：	LAB9
		發佈日期：	2022/05/16

三. 輸入及輸出結果：

 D:\Program\C&CPP\LAB9\LAB9.exe

```

Input number of vertex and edges : 8 10
Input the edge-0 : 0 1
Input the edge-1 : 2 0
Input the edge-2 : 2 3
Input the edge-3 : 1 3
Input the edge-4 : 3 4
Input the edge-5 : 3 5
Input the edge-6 : 4 6
Input the edge-7 : 4 7
Input the edge-8 : 6 5
Input the edge-9 : 7 5
Path : 2 0 1 3 4 6 7 5

```

四. 心得與討論：

這次的實作題目要求不可使用遞迴的方式進行，因此花了蠻多時間思考該如何撰寫程式碼，後續也花了不少時間 Debug；最後有些小問題無法解決，因此也參考老師提供程式碼下去思考，將老師的程式碼看過兩遍後，也迅速地發現自己的盲點並將其修正才成功解出此題。

透過這次的實作，也大致理解遞迴轉非遞迴的撰寫方式，但仍需要多加嘗試才能撰寫成功，因此實作後的收穫也非常多。