

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

# LAB8

## 圖論與其資料結構應用

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：2022/05/09

版本校定紀錄：

版本	更新紀錄	發佈日期
0.0.0.0	初版完成	2022/05/09

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

## 一. 實驗需求：

### (一). 題目說明

#### 描述：

在地圖上有  $n$  個藏寶點，每個藏寶點有若干價值的寶物，且只要能到達藏寶點一定可以取得該藏寶點的寶藏。地圖上一共有  $m$  條道路，每條道路連接兩個藏寶點，而且每條道路都是雙向可以通行的。一開始可以自由選擇落下的位置，因此必須決定要從哪裡開始才可以獲得最多的寶藏總價值。

#### 輸入說明

第一行是兩個正整數  $n$  與  $m$  分別代表藏寶地點數與道路數，地點會以  $0 \sim n-1$  編號，第二行  $n$  個非負整數，依序是每一個地點的寶藏價值，每個地點的寶藏價值不超過 100。接下來有  $m$  行，每一行兩個整數  $a$  與  $b$  代表一個道路連接的兩個地點編號。且  $n$  不超過  $5 \times 10^4$ ，而  $m$  不超過  $5 \times 10^5$ 。兩點之間可能有多條道路，有些道路的兩端點可能會是同一地點。

#### 輸出說明：

每筆測試資料輸出一行最大可以獲得的寶藏總價值的答案。

### (二). 演算法

#### i. 虛擬碼

定義 `maxVertex` 並設定大小為頂點最大數量 50000；

宣告三個整數(`int`)型態變數 `n`, `m`, `maxValue`，分別用於存放測資頂點數量、相連道路數量及路線最大的寶藏價值；

宣告一個整數(`int`)型態陣列變數 `value`，大小為 `maxVertex`，用於存放每個頂點的寶藏價值；

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

宣告一個 `vector` 型態變數 `roads` 並指定存放整數(`int`)型態的資料，大小為 `maxVertex`，用於存放頂點與其相連頂點的資料；

宣告一個布林(`boolean`)型態陣列變數 `haveVisit`，大小為 `maxVertex`，紀錄每個頂點是否被造訪過，並將初始值設為 `false`；

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

```

int main(){
    輸出提示訊息「請輸入頂點數及道路數」；
    輸入該筆測資頂點數及道路數，並存入 n 和 m 中；

    輸出提示訊息「請輸入每個頂點的寶藏價值」；
    for i = 0 ~ i < m (for i++){
        輸入第 i 個頂點的寶藏價值，並存入 value[i] 中；
    }

    for i = 0 ~ i < n (for i++){
        輸出提示訊息「請輸入第 i 條道路」；
        宣告兩個整數(int)變數 a 和 b，存放道路兩端的頂點；
        輸入道路兩端頂點，並存入 a 和 b 中；
        在 roads[a] 位置存入 b，代表 a 到 b 有一條道路；
        在 roads[b] 位置存入 a，代表 b 到 a 有一條道路；
    }

    因探索尚未開始，因此將 maxValue 設為 0；

    for v = 0 ~ v < m (for v++){
        如果 haveVisit[v] 為 false，代表頂點 v 未被造訪{
            haveVisit[v] 設為 true，代表已被拜訪；
            宣告一個整數型態變數 returnValue；
            將 visit(v) 回傳的值存入 returnValue；
            // 「visit(v)」將回傳頂點 v 後續探索得到的寶藏價值
            如果 returnValue 大於 maxValue 代表 v 探索的價值更高{
                將 maxValue 設為 returnValue；
            }
        }
    }

    輸出 maxValue；

    return 0;
}

```

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

```

宣告一個會回傳整數的函數 visit，並需傳入一個參數 v{
    宣告一個整數(int)型態變數 nowValue，並存入 value[v]；
    //nowValue 將存放此頂點及後續探索頂點的寶藏價值總和

    for v_next : roads[v] 檢查 v 後續的頂點{
        如果 haveVisit[v_next]為 false，代表 v_next 未被造訪{
            將 haveVisit[v_next]設為 true；
            nowValue 設為 nowValue 加 visit(v_next)；
        }
    }
    return nowValue；
    //回傳頂點 v 及其後續能探索的頂點的寶藏價值總和；
}

```

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

## 二. 完整程式碼：

```

#include<bits/stdc++.h>
#define maxVertex 50000
// 頂點數不超過  $5 \times 10^4$ ，因此定義一個最大頂點數
int n, m, maxValue, value[maxVertex];
// n 為該測資的頂點數；m 為該測資的道路數
// maxValue 用於存放目前所有路線最大的寶藏價值
// value 存放每個頂點的寶藏價值
using namespace std;
vector<int> roads[maxVertex];
// 宣告一個 vector 陣列型態的整數陣列，存放頂點與其他頂點相連的資料
bool haveVisit[maxVertex] = {false};
// 紀錄該頂點是否被訪問過，如果有就不再重複訪問，以免重複計算寶藏價值

int visit(int v){
// 進行 DFS 訪問動作的 function，傳入本此要訪問的頂點 v
    int nowValue = value[v];
    // 記錄此頂點及後續由該頂點延伸所獲得的寶藏價值
    for(int v_next : roads[v]){
// 將與頂點 v 相連的其他頂點依序放入 v_next
        if(!haveVisit[v_next]){
// 如果 v_next 尚未被訪問，則進入 if 內
            haveVisit[v_next] = true;
            nowValue += visit(v_next);
// 將 v_next 所能獲得的寶藏價值加入頂點 v 目前的寶藏價值
        }
    }
    return nowValue;
// 回傳頂點 v 所能獲得寶藏價值
}

int main(){
    printf("\n Input number of vertex and road : ");
    scanf("%d%d", &m, &n);
    // 輸入本次測資的頂點數及道路數

```

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

```

printf("\n Input value of vertex : ");
for(int i = 0 ; i < m ; i++){
    scanf("%d", &value[i]);
} // 輸入每個頂點寶藏的價值

for(int i = 0 ; i < n ; i++){
    printf("\n Input road%d : ", (i + 1));
    int a, b;
    scanf("%d%d", &a, &b);
    // 輸入道路起點與終點
    roads[a].push_back(b);
    roads[b].push_back(a);
    // 因為本題為無向圖，因此道路為雙向通行，故兩邊都需紀錄連接
} // 輸入所有相連的頂點，也就是道路

maxValue = 0;
// 因為還沒開始探索，因此將所探索路線的最大寶藏價值設為 0
for(int v = 0 ; v < m ; v++){
    // 用迴圈搜尋所有頂點
    if(!haveVisit[v]){
        // 如果頂點未被訪問，就進入 if 內
        haveVisit[v] = true;
        // 將該頂點設為已被訪問
        int returnValue = visit(v);
        // 透過 visit 獲得該節點往下探索後，所能獲得的寶藏價值
        if(returnValue > maxValue){
            // 如果回傳的價值比目前的寶藏價值大，進入 if 內
            maxValue = returnValue;
            // 將目前最大寶藏價值設為回傳的寶藏價值
        }
    }
}

printf("\n The max value is %d.", maxValue);
// 輸出最大寶藏價值
return 0;
}

```

資訊工程學系	資料結構應用	文件編號：	LAB8
		發佈日期：	2022/05/09

### 三. 輸入及輸出結果：

```

Input number of vertex and road : 7 6
Input value of vertex : 5 2 4 2 1 1 8
Input road1 : 5 1
Input road2 : 1 3
Input road3 : 1 4
Input road4 : 2 0
Input road5 : 2 0
Input road6 : 3 3
The max value is 9.

```

```

Input number of vertex and road : 3 2
Input value of vertex : 2 1 5
Input road1 : 1 0
Input road2 : 0 1
The max value is 5.

```

### 四. 心得與討論：

本次實作的資料結構「圖」，因為有上一次 LAB7 實作練習打基底，因此在實作上理解的速度快了不少；也多虧了上次的練習及理解，這次大概在理解題意之後，就只耗費 15 分鐘左右將題目要求的功能完成。

這次實作程式碼的部分，能發現自己經過這十幾周的時間，已能快速地將 C/C++ 的程式碼撰寫出來，與 LAB1 及 LAB2 時相比快了不少；也相信未來自己能夠更精熟 C 和 C++ 的語法，並在未來的專題或競賽上有更好的發揮。