

# Data Structure Homework 3

0416303 楊博凱 資工資電

這次作業是要實作建構 Minimum Cost Spanning Tree 的演算法，首先要先了解演算法的步驟。一開始，我寫了兩個 class，class group：代表由兩個 node 與一個 connection 成的組合；class MCST：代表一個 Minimum Cost Spanning Tree。要建立一個 MCST 首先呼叫 member function build，build 會先將輸入的 group array 利用 Quick Sort 做排序，使其依照 cost 的大小由小到大排列。再來，將 group 從 array 前方依序取出，由 is\_circle function 判斷是否會與 MCST 內的 node 構成環；若無環產生，則呼叫 insert function 將此 group 加入 MCST 中。

```
void build(group<Type> *input_arr,int how_much)
{
    quick_sort_group(input_arr,0,how_much-1); // First, sort the input : group array
    for(int i=0;i<how_much;i++){
        // For all groups in sorted array, take out groups inorder and determine if it will build a circle with the groups in MCST ( Minimum Cost Spanning Tree )
        if(!is_circle(input_arr[i]))
            insert(input_arr[i]); // if the group won't cause a circle, insert it
    }
}
```

在 is\_circle function 中，首先我建立了一個二維陣列 same，用來判斷不同 link 中分別與 input group 的 number1 和 number2 重複的個數。Initial value 均設為零，若 link X 與 input group 的 number1 有相同數字，則 same[0][X]加一；反之，若 link X 與 input group 的 number2 有相同數字，則 same[1][X]加一。最後，若有其中一條 link Y 的 same[0][Y]與 same[1][Y]均大於等於 1，則表示此 input group 會在 link Y 上造成 circle(環)的產生。

```
bool is_circle(group<Type> in) // to consider if adding this new group will cause a circle
{
    int same[2][field_num]; // array same consider field x has how many same numbers
    for(int i=0;i<field_num;i++){
        same[0][i]=0; // initial set to zero ( each field has zero same numbers )
        same[1][i]=0;
    }
    for(int i=0;i<num;i++){
        if(arr[i].num1==in.num1||arr[i].num2==in.num1)
            same[0][arr[i].field]++; // if has same number then add same[ field number ] for once
        else if(arr[i].num1==in.num2||arr[i].num2==in.num2)
            same[1][arr[i].field]++;
    }
    for(int i=0;i<field_num;i++){
        if(same[0][i]>=1&&same[1][i]>=1) // has over two same numbers in the link ( will cause a circle )
            return true; // has a circle
    }
    return false; // no circle
}
```

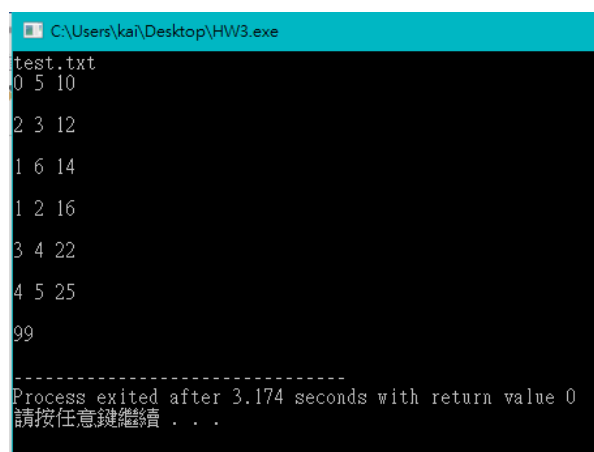
在 insert function 中，首先判斷 Minimum Cost Spanning Tree 的元素個數是否超出其 maximum 範圍，若超出，則 double 其空間。因為在 insert 前我們已經了解到此 group 不會造成新的環，因此可以放心加入他；又我們已經將 array 排序，可以確定它們是按照 cost 大小由小到大排入。為了判斷此 group 屬於 MCST 中哪一條 link，我們在 class group 中加入了一個 field 值來判斷每個 group 屬於哪一條 link。若新加入的 group 與 X link 中的 group 有相同的 node number，則此新的 group 屬於 link X；但若 group 又與 Y link 中的 group 有相同的 node number，則他連接了兩條 link，所以我們也必須更改其中一條 link 中 groups 的 field number。

```
void insert(group<Type> in) // insert a group
{
    if(num+1==max){ // if Minimum Cost Spanning Tree is full, double the size of it
        max*=2;
        group<Type> *temp_arr=new group<Type> [max];

        for(int i=0;i<num;i++){
            temp_arr[i]=arr[i];
        }
        delete arr;
        arr=temp_arr;
    }
    arr[num]=in; // add group in Minimum Cost Spanning Tree
    for(int i=0;i<num;i++){
        // Find which link (field) does the new group belongs to
        // If the numbers of the new group is same as the number of a old group, they have the same link (field)
        if(arr[i].num1==arr[num].num1||arr[i].num1==arr[num].num2||arr[i].num2==arr[num].num1||arr[i].num2==arr[num].num2){
            if(arr[num].field==0) // the new group doesn't belongs to any link (field) before, so it belongs to this link (field)
                arr[num].field=arr[i].field;
            else{ // the new group belongs to a link (field) before, so we union two links together
                for(int j=0;j<num;j++){
                    if(arr[j].field==arr[i].field)
                        arr[j].field=arr[num].field;
                }
            }
        }
    }
    if(arr[num].field==0) // the new group doesn't belongs to any link (field) before, and it doesn't has any semi-number in MCST, so it belongs to a new link
        arr[num++].field=field_num++;
    else // the new group belongs to a link (field) before
        num++;
    return;
}
```

在這次作業中所用的 array，我採用 dynamic 的方式分配空間，initial space 設成 2，若超過範圍，則 double 其空間。至於其他與本次作業比較不相關的 function ( 如 Quick Sort , print , set 與 constructor 等等 ) 或是 main 中的實作過程，則在程式的註解中再詳細敘述。

### 助教測資 Demo：



```
C:\Users\kai\Desktop\HW3.exe
test.txt
0 5 10

2 3 12

1 6 14

1 2 16

3 4 22

4 5 25

99

-----
Process exited after 3.174 seconds with return value 0
請按任意鍵繼續...
```