

Kruskal's Algorithm

Description

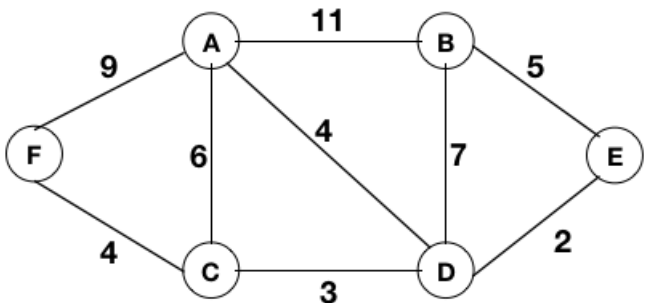
實作Kruskal's Algorithm最小生成樹

給出一個n個結點的無向圖，找最大邊減去最小邊的最小生成樹。

圖中不含重邊和自環。

範例參考圖

A-F由數字表示



Input

第一列輸入幾個節點 和幾條邊

輸入兩個節點表示指定邊 下一步輸入其權重

以上過程結束條件為邊權輸入完成

再來持續回到上一部輸入節點數量和邊

直到輸入0 0結束

Output

該最小生成樹的最大邊減去最小邊的數值

Sample Input 1

```
6 9
4 5 2
3 4 3
1 4 4
3 6 4
2 5 5
1 3 6
2 4 7
1 6 9
1 2 11
0 0
```

Sample Output 1

```
3
```

Problems

Announcements

Submissions

Rankings

View Contest

Information

ID 122101

Time Limit 3000MS

Memory Limit 512MB

IO Mode Standard IO

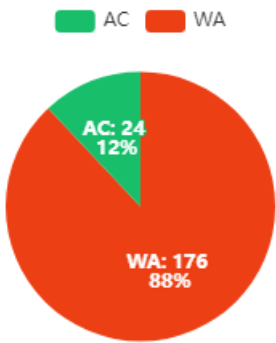
Created By m0968390

Level Low

Tags [Show](#)

Statistic

Details



Language: C



Theme: Solarized Light

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int SIZE;
4 struct edge{
5     int w;
6     int p1;
7     int p2;
8 }V[100];
9 int matrix[100][100];
10 int vset[100]; //set, damn
11 void kruskal(void);
12 int v,e;
13 int Result;
14 int main(){
15     int I,J,W;
16     int i,j;
17     scanf("%d %d",&v,&e);
18     int first=1;
19     while(v!=0&&e!=0){
20         if(first==0)printf("\n");
21         SIZE=v;
22         for(i=0;i<SIZE;i++){
23             V[i].w=0;
```

```
27         matrix[i][j]=-1;
28     }
29 }
30 for(i=0;i<e;i++){
31     scanf("%d %d %d",&I,&J,&W);
32     V[i].p1=I-1;
33     V[i].p2=J-1;
34     V[i].w=W;
35 }
36 kruskal();
37 printf("%d",Result);
38 first=0;
39 scanf("%d %d",&v,&e);
40 }
41
42 /*for(i=0;i<SIZE;i++){
43     for(j=0;j<SIZE;j++){
44         printf("%d\t",matrix[i][j]);
45     }
46     printf("\n");
47 }*/
48
49 // 結束輸入，輸出結果
```

✔ You have solved the problem

Submit