

# darkRoads\_MST.java

```
1 /* MST Accepted Solution UVa 11631 : Dark Roads
2  * Minimum Spanning Tree (MST) Problem to calculate the least
3  * amount of road length to
4  * be lighted up.
5  * */
6
7 import java.util.*;
8
9 class darkRoads_MST{
10     static int M;
11     static int N;
12     static PriorityQueue<Edge> E;
13
14     static class DisjointSet{
15         int[] pset;
16         public DisjointSet(){
17             pset = new int[N];
18             for(int i = 0; i<pset.length; i++){
19                 pset[i] = i;
20             }
21         }
22
23         public int findSet(int i){
24             return pset[i]==i ? pset[i] : (pset[i] =
25 findSet(pset[i]));
26         }
27
28         public void unionSet(int i, int j){
29             pset[findSet(i)] = findSet(j);
30         }
31
32         public boolean isSameSet(int i, int j){
33             return (findSet(i)==findSet(j));
34         }
35     }
36
37     static class Edge implements Comparable<Edge> {
38         int to, weight, from;
39         public Edge(int f, int t, int w){
40             to=t;
41             weight = w;
42             from = f;
43         }
44         @Override
```

darkRoads\_MST.java

```
39     public int compareTo(Edge e) {
40         return this.weight - e.weight;
41     }
42 }
43
44 public static void addEdge(int u, int v, int w) {
45     E.add(new Edge(u, v, w));
46 }
47 public static void init_MSTkruskal() {
48     E = new PriorityQueue<Edge>();
49 }
50 public static int MSTkruskal(DisjointSet ds) {
51     int cost = 0;
52     while(!E.isEmpty()) {
53         Edge e = E.poll();
54         if(!(ds.isSameSet(e.from, e.to))) {
55             ds.unionSet(e.from, e.to);
56             cost+=e.weight;
57         }
58     }
59     return cost;
60 }
61
62 public static void main(String[] args) {
63     Scanner scan = new Scanner(System.in);
64     while(true) {
65         init_MSTkruskal();
66         int w = -1;
67         int total = 0;
68         N = scan.nextInt();
69         M = scan.nextInt();
70         if(M==0 && N == 0)
71             break;
72 //         makeGraph(N);
73         for(int i = 0; i<M; i++) {
74
75             addEdge(scan.nextInt(), scan.nextInt(), w = scan.nextInt());
76             total+=w;
77         }
78         DisjointSet ds = new DisjointSet();
```

darkRoads\_MST.java

```
78         System.out.println(total - MSTkruskal(ds));
79     }
80     scan.close();
81 }
82 }
83
```