

```

1 import java.util.*;
2
3 /** @author Amar Bessedik
4  * This class designs an Minimum Spanning Tree (or MST).
5  * An MST is subgraph with minimum weight & without cycles of a connected graph.
6  */
7 public class MinimumSpanningTree
8 {
9
10     private ArrayList<Edge> mst;//list that grows as edges are added.
11     private int weight;//total weight
12     private int condition;//Kruskal's algo stops as this condition is satisfied.
13     private int V;//# of vertices of G. Needed as an MST must have (V - 1) edges.
14
15     /**
16      * Constructor
17      *
18      * @param G is a graph for which Kruskal finds its MST if it exists.
19      */
20     public MinimumSpanningTree(Graph G)
21     {
22         this.mst = new ArrayList<>();
23         this.V = G.getVertices();
24         this.condition = V - 1;
25         this.weight = 0;
26     }// end constructor
27
28     /**
29      * @param weight updates the weight of the MST.
30      */
31     public void update(int weight)
32     {
33         this.weight += weight;
34     }//end update
35
36     /**
37      * @param e add e to MST
38      */
39     public void add(Edge e)
40     {
41         mst.add(e);
42     }//end add
43
44     /**
45      * @return size of MST.
46      */
47     public int size()
48     {
49         return mst.size();
50     }//end size
51

```

```

52     /**
53      * @return true if there are (V - 1) edges in the MST.
54      */
55     public boolean satisfied()
56     {
57         return size() == condition;
58     }
59
60     /**
61      * Shows results after execution of the Kruskal's function.
62      *
63      * @return a message depending on whether there is an MST or not.
64      */
65     public void output()
66     {
67         if (!satisfied())
68             System.out.println("THE GRAPH IS NOT CONNECTED! No MST.\n");
69
70         else
71         {
72             String mstMessage = "\nMINIMUM SPANNING TREE:\n\n";
73             mstMessage += String.format("%s %4s %24s", "V1", "V2",
74                                     "W\n-----\n");
75             for (Edge e : mst)
76                 mstMessage += e + "\n";
77             mstMessage += "\nTOTAL WEIGHT: " + weight + "\n\n";
78
79             System.out.println(mstMessage);
80         }
81     } //end mst_output
82 } //end MinimumSpanningTree class
83

```