```java
/**
 * @author Amar Bessedik Designs the Kruskal's algorithm for finding minimum
 * spanning trees of graphs.
 */
public class Kruskal
{
   private MinimumSpanningTree mst;// MST object, to potentially hold (V - 1) edges.
   private DisjointSet ds;//To hold edges each in a disjoint set.
   private HeapSort hs;//Needed to sort the edges of a graph.
   private int N;//Number of vertices
   private int u, v, wt, u_set, v_set;//Edge params: vertex1, vertex2, weight & sets.

   //Constructor
   public Kruskal(Graph G)
   {
      this.N = G.getVertices();// # vertices
      this.hs = new HeapSort(); // heapSort instance
      this.ds = new DisjointSet(N);//Disjoint set of capacity N.
      this.mst = new MinimumSpanningTree(G);// MST instance.
   }//end constructor

   /**
    * Kruskal's function.
    * @param V # number of vertices.
    * @param E array of graph edges.
    */
   public void kruskal(int V, Edge[] E)
   {
      Edge e;//Shortest edge yet to be considered.
      int count = 0;//counter of the graph's edges.
      int n = E.length;//# of edges.

      //Sort Edges in inceasinding order of weight.
      hs.heapSort(E, n);

      while ((count < n) && !mst.satisfied())
      {
         e = E[count++];//Shortest edge yet to consider.
         get_parameters(e);//Get the edge's parameters.

         u_set = ds.find2(u);//The label of vertex u.
         v_set = ds.find2(v);//The label of vertex v.

         //If adding the edge to the MST would create a cycle.
         if (u_set == v_set)
            continue;

         //Otherwise - NO CYCLE
         ds.merge(u_set, v_set); // merge the sets into one disjoint set.
         mst.add(e);// Add edge to MST
         mst.update(wt);// update total weight.

      }//end while
      //Show results according to weither there is an MST or not.
      mst.output();
   }//end kruskal

   /**
```

```
59      * @param e gets vertices and weight of e.
60      */
61     private void get_parameters(Edge e)
62     {
63        this.u = e.getVertex1();// get first vertex
64        this.v = e.getVertex2();// get second vertex.
65        this.wt = e.getWeight();// get edge's weight.
66     }//end extract_parameters
67 }//end Kruskal's Class
68
```