Design & Analysis of Algorithm

Name – Jay Lapani

Roll No. - 21BCP150

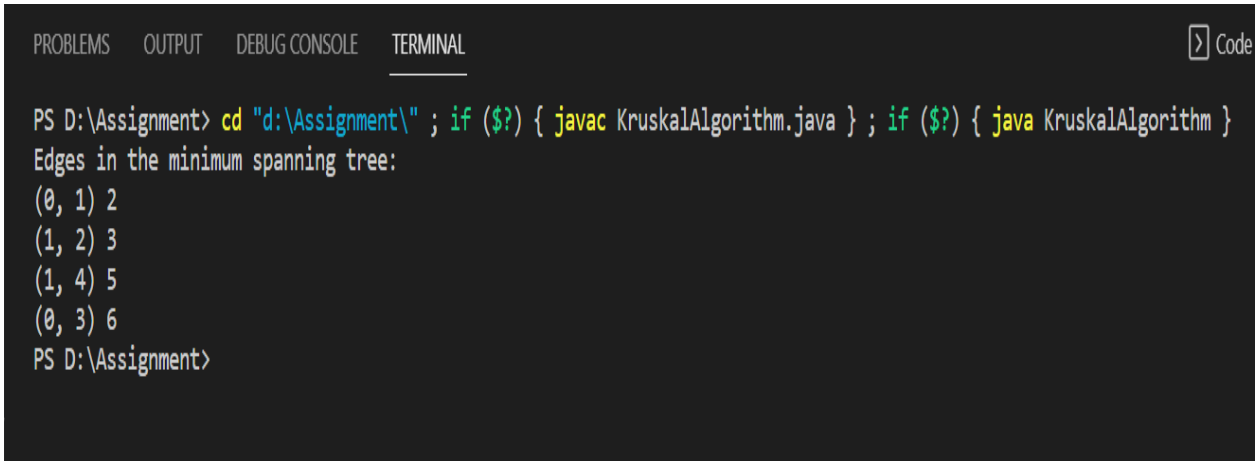
Div - 3 , G - 5

Greedy Algorithm

```
import java.util.*;
public class KruskalAlgorithm {

 static class Edge {
 int src, dest, weight;

 public Edge(int src, int dest, int weight) {
 this.src = src;
 this.dest = dest;
 this.weight = weight;
 }
 }

 static class Subset {
 int parent, rank;

 public Subset(int parent, int rank) {
 this.parent = parent;
 this.rank = rank;
 }
 }

 static Edge[] kruskalMST(Edge[] edges, int
numVertices) {
 Arrays.sort(edges, Comparator.comparingInt(e->
e.weight));

 Subset[] subsets = new Subset[numVertices];
 for (int i = 0; i < numVertices; i++) {
 subsets[i] = new Subset(i, 0);
 }

 Edge[] result = new Edge[numVertices - 1];
 int index = 0;

 for (int i = 0; i < edges.length && index <
numVertices - 1; i++) {
 Edge edge = edges[i];
```

```
 // Find the parent of the source and destination
vertices
 int srcParent = find(subsets, edge.src);
 int destParent = find(subsets,
edge.dest);

 if (srcParent != destParent) {
 result[index++] = edge;
 union(subsets, srcParent,
destParent);
 }
 }

 return result;
 }

 static int find(Subset[] subsets, int i) {
 if (subsets[i].parent != i) {
 subsets[i].parent = find(subsets,
subsets[i].parent);
 }
 return subsets[i].parent;
 }

 static void union(Subset[] subsets, int x, int y)
{
 int xroot = find(subsets, x);
 int yroot = find(subsets, y);

 if (subsets[xroot].rank <
subsets[yroot].rank) {
 subsets[xroot].parent = yroot;
 } else if (subsets[xroot].rank >
subsets[yroot].rank) {
 subsets[yroot].parent = xroot;
 } else {
 subsets[yroot].parent = xroot;
```

```
 subsets[xroot].rank++;
 }
 }

 public static void main(String[] args) {
 int numVertices = 5;
 Edge[] edges = {
 new Edge(0, 1, 2),
 new Edge(0, 3, 6),
 new Edge(1, 2, 3),
 new Edge(1, 3, 8),
 new Edge(1, 4, 5),
 new Edge(2, 4, 7),
 new Edge(3, 4, 9)
 };

 Edge[] mst = kruskalMST(edges, numVertices);

 System.out.println("Edges in the minimum spanning
tree:");
 for (Edge edge : mst) {
 System.out.printf("(%d, %d) %d\n",
edge.src, edge.dest, edge.weight);
 }
 }
}
```


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Code
PS D:\Assignment> cd "d:\Assignment\" ; if ($?) { javac KruskalAlgorithm.java } ; if ($?) { java KruskalAlgorithm }
Edges in the minimum spanning tree:
(0, 1) 2
(1, 2) 3
(1, 4) 5
(0, 3) 6
PS D:\Assignment>