using System;

using System.Collections.Generic;

using System.Linq;

class KruskalAlgorithm

{

class Edge

{

public int Source { get; set; }

public int Destination { get; set; }

public double Weight { get; set; }

}

static int Parent(int[] parent, int i)

{

if (parent[i] != i)

parent[i] = Parent(parent, parent[i]);

return parent[i];

}

static void SvUnion(int[] parent, int[] rank, int x, int y)

{

int xRoot = Parent(parent, x);

int yRoot = Parent(parent, y);

if (rank[xRoot] < rank[yRoot])

parent[xRoot] = yRoot;

else if (rank[xRoot] > rank[yRoot])

parent[yRoot] = xRoot;

else

{

parent[yRoot] = xRoot;

rank[xRoot]++;

}

}

static void KruskalAlg(List<Edge> edges, int V)

{

edges = edges.OrderBy(edge => edge.Weight).ToList();

List<Edge> result = new List<Edge>();

int[] parent = new int[V];

int[] rank = new int[V];

for (int j = 0; j < V; j++)

{

parent[j] = j;

rank[j] = 0;

}

int e = 0;

int i = 0;

while (e < V - 1)

{

Edge nextEdge = edges[i++];

int x = Parent(parent, nextEdge.Source);

int y = Parent(parent, nextEdge.Destination);

if (x != y)

{

result.Add(nextEdge);

SvUnion(parent, rank, x, y);

e++;

}

}

double WeightOfMST = 0;

Console.WriteLine("Минимальное Остовное Дерево(МОД): ");

Console.WriteLine("Ребро \t Вес");

foreach (Edge edge in result)

{

Console.WriteLine($"{edge.Source} - {edge.Destination}: \t {edge.Weight}");

WeightOfMST += edge.Weight;

}

Console.WriteLine("Общая длина МОД: " + WeightOfMST);

}

public static void Main()

{

// Console.Write("Введите кол-во вершин графа: ");

int v = 8; //Convert.ToInt32(Console.ReadLine());

List<Edge> edges = new List<Edge>();

/\* while (true)

{

int a; int b; double c; string d;

Console.WriteLine("Введите длину ребра(если закончили ввод ребер, нажмите <0>, <-> или Enter: ");

d = Console.ReadLine();

if (d == "" || d == "0" || d == "-") break;

c = Convert.ToDouble(d);

Console.WriteLine("Введите номер первой вершины ребра: ");

a = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Введите номер второй вершины ребра: ");

b = Convert.ToInt32(Console.ReadLine());

edges.Add(new Edge { Source = a - 1, Destination = b - 1, Weight = c });

}\*/

edges.Add(new Edge { Source = 0, Destination = 1, Weight = 2 });

edges.Add(new Edge { Source = 0, Destination = 2, Weight = 2 });

edges.Add(new Edge { Source = 0, Destination = 4, Weight = 7 });

edges.Add(new Edge { Source = 2, Destination = 3, Weight = 9 });

edges.Add(new Edge { Source = 2, Destination = 4, Weight = 2 });

edges.Add(new Edge { Source = 2, Destination = 2, Weight = 2 });

edges.Add(new Edge { Source = 2, Destination = 4, Weight = 5 });

edges.Add(new Edge { Source = 3, Destination = 4, Weight = 10 });

edges.Add(new Edge { Source = 3, Destination = 6, Weight = 2 });

edges.Add(new Edge { Source = 3, Destination = 5, Weight = 5 });

edges.Add(new Edge { Source = 3, Destination = 7, Weight = 16 });

edges.Add(new Edge { Source = 4, Destination = 6, Weight = 9 });

edges.Add(new Edge { Source = 5, Destination = 7, Weight = 17 });

edges.Add(new Edge { Source = 6, Destination = 7, Weight = 17 });

Console.Clear();

KruskalAlg(edges, v);

}

}