***Problem Defination:-***

Implementation of Randomized Minimum Cost Spanning Tree Algorithm to find nearest neighbour in range from the current location. Use graph input with distances.

***Detailed Description:-***

An undirected weighted graph was taken as input and then all vertices are sorted in increasing order then for each vertices we find cheapest edge that connects it to other vertice.After that we will be having n subtree containing different components.Then we have to combine this components to form final Randomized Minimum Cost Spanning Tree.

***Technology Stack: -***

The entire coding is done in Java with NetBeans IDE 8.0.1.

***Classes and Functions: -***

The program makes the use of 2 classes: -

1. Rmst (Main Class)

2. Graph

**1.Rmst: -**

This is the main class where we take input as V Vertices and E Edges

And then we make call to graph constructor this will create a graph with V

Vertices and E edges .Then RandomizedMST function gives final output as

Randomized minimum cost spanning tree.

**2.Graph: -**

This class contains two other classes named as Edge to represent weighted edge

And other as Subset to represent a subset for union-find. A constructor as Graph

Which represents a connected,weighted and undirected graph as collection of

Edges. The function **RandomizedMST()**  takes created graph as input and for

Each vertice of graph find a nearest vertices with minimum edge cost.

Initially there will be V different trees and we need to combine this subtrees

Until all subtrees are not combined into a single a MST.

The function **find()** is utility function to find set of an element and the function

**Union()** does the union of two sets x and y.