**DESIGN ANALYSIS AND ALGORITHMS**

**LAB ASSIGNMENT – 1**

**NAME : PRATHAPANI SATWIKA**

**REG.NO. : 20BCD7160**

**1)** **Write a code for Kruskal’s algorithm using java**

**CODE :**

**OUTPUT :**

**2) Write a code for Prim’s algorithm using java**

**CODE :**

**package** Lab1b;

**import** java.util.Arrays;

**public** **class** Lab1b {

**public** **void** Prim(**int** G[][], **int** V) {

**int** INF = 9999999;

**int** no\_edge; // number of edge

// create a array to track selected vertex

// selected will become true otherwise false

**boolean**[] selected = **new** **boolean**[V];

// set selected false initially

Arrays.*fill*(selected, **false**);

// set number of edge to 0

no\_edge = 0;

// the number of egde in minimum spanning tree will be

// always less than (V -1), where V is number of vertices in

// graph

// choose 0th vertex and make it true

selected[0] = **true**;

// print for edge and weight

System.***out***.println("Edge : Weight");

**while** (no\_edge < V - 1) {

// For every vertex in the set S, find the all adjacent vertices

// , calculate the distance from the vertex selected at step 1.

// if the vertex is already in the set S, discard it otherwise

// choose another vertex nearest to selected vertex at step 1.

**int** min = INF;

**int** x = 0; // row number

**int** y = 0; // col number

**for** (**int** i = 0; i < V; i++) {

**if** (selected[i] == **true**) {

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

// not in selected and there is an edge

**if** (!selected[j] && G[i][j] != 0) {

**if** (min > G[i][j]) {

min = G[i][j];

x = i;

y = j;

}

}

}

}

}

System.***out***.println(x + " - " + y + " : " + G[x][y]);

selected[y] = **true**;

no\_edge++;

}

}

**public** **static** **void** main(String[] args) {

Lab1b g = **new** Lab1b();

// number of vertices in grapj

**int** V = 5;

// create a 2d array of size 5x5

// for adjacency matrix to represent graph

**int**[][] G = { { 0, 9, 75, 0, 0 }, { 9, 0, 95, 19, 42 }, { 75, 95, 0, 51, 66 }, { 0, 19, 51, 0, 31 },

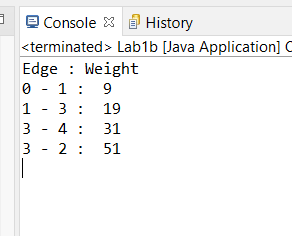
{ 0, 42, 66, 31, 0 } };

g.Prim(G, V);

}

}

**OUTPUT :**

****