**Kruskal's algorithm**

**Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.**

#include <stdio.h>

int i, j, k, a, b, u, v, n, ne = 1;

int min, mincost = 0, cost[10][10], parent[10];

int find(int);

int uni(int, int);

void main() {

printf("\n\tImplementation of Kruskal's Algorithm\n");

printf("\nEnter the no. of vertices:");

scanf("%d", & n);

printf("\nEnter the cost adjacency matrix:\n");

for (i = 1; i <= n; i++) {

for (j = 1; j <= n; j++) {

scanf("%d", & cost[i][j]);

if (cost[i][j] == 0)

cost[i][j] = 999;

}

}

printf("The edges of Minimum Cost Spanning Tree are\n");

while (ne < n) {

for (i = 1, min = 999; i <= n; i++) {

for (j = 1; j <= n; j++) {

if (cost[i][j] < min) {

min = cost[i][j];

a = u = i;

b = v = j;

}

}

}

u = find(u);

v = find(v);

if (uni(u, v)) {

printf("%d edge (%d,%d) =%d\n", ne++, a, b, min);

mincost += min;

}

cost[a][b] = cost[b][a] = 999;

}

printf("\n\tMinimum cost = %d\n", mincost);

}

int find(int i) {

while (parent[i])

i = parent[i];

return i;

}

int uni(int i, int j) {

if (i != j) {

parent[j] = i;

return 1;

}

return 0;

}

**OUTPUT:**

Implementation of Kruskal's Algorithm

Enter the no. of vertices: 6

Enter the cost adjacency matrix:

0 3 10 10 6 5

3 0 1 10 10 4

10 1 0 6 10 4

10 10 6 0 8 5

6 10 10 8 0 2

5 4 4 5 2 0

The edges of Minimum Cost Spanning Tree are

1 edge (2,3) =1

2 edge (5,6) =2

3 edge (1,2) =3

4 edge (2,6) =4

5 edge (4,6) =5

Minimum cost = 15