

## PRIMS

```
import java.util.Scanner;
public class Prims {
    final static int MAX = 20;
    static int n;
    static int cost[][];
    static Scanner scan = new Scanner(System.in);
    public static void main(String[] args) {
        ReadMatrix();
        Prims();
    }
    static void ReadMatrix() {
        int i, j;
        cost = new int[MAX][MAX];
        System.out.println("Enter the number of nodes:");
        n = scan.nextInt();
        System.out.println("Enter the adjacency matrix:");
        for (i = 1; i <= n; i++) {
            for (j = 1; j <= n; j++) {
                cost[i][j] = scan.nextInt();
                if (cost[i][j] == 0) {
                    cost[i][j] = 999;
                }
            }
        }
    }
    static void Prims() {
        int visited[] = new int[10];
        int ne = 1, i, j, min, a = 0, b = 0, u = 0, v = 0;
        int mincost = 0;
        visited[1] = 1;
        while (ne <= n) {
            for (i = 1, min = 999; i <= n; i++) {
                for (j = 1; j <= n; j++) {
                    if (cost[i][j] < min) {
                        if (visited[i] != 0) {
                            min = cost[i][j];
                            a = u = i;
                            b = v = j;
                        }
                    }
                }
            }
            if (visited[u] == 0 || visited[v] == 0) {
                System.out.println("Edge"+ne+":(" + a + ", " + b + ") " + " cost : " + min);
                mincost += min;
                visited[b] = 1;
            }
            cost[a][b] = cost[b][a] = 999;
            ne++;
        }
        System.out.println("Minimun cost: " + mincost);
    }
}
```

## KRUSKALS

```
import java.util.Scanner;
public class Kruskals {
    final static int MAX = 20;
    static int n;
    static int cost[][];
    static Scanner scan = new Scanner(System.in);
    public static void main(String[] args) {
        ReadMatrix();
        Kruskals();
    }
    static void ReadMatrix() {
        int i, j;
        cost = new int[MAX][MAX];
        System.out.println("Implementation of Kruskal's algorithm");
        System.out.println("Enter the no. of vertices");
        n = scan.nextInt();
        System.out.println("Enter the cost adjacency matrix");
        for (i = 1; i <= n; i++) {
            for (j = 1; j <= n; j++) {
                cost[i][j] = scan.nextInt();
                if (cost[i][j] == 0) {
                    cost[i][j] = 999;
                }
            }
        }
    }
    static void Kruskals() {
        int a = 0, b = 0, u = 0, v = 0, i, j, ne = 1, min, mincost = 0;
        int parent[] = new int[9];
        for (i = 1; i <= n; i++) {
            parent[i] = 0;
        }
        System.out.println("The edges of Minimum Cost Spanning Tree are");
        while (ne < n) {
            min = 999;
            for (i = 1; i <= n; i++) {
                for (j = 1; j <= n; j++) {
                    if (cost[i][j] < min) {
                        min = cost[i][j];
                        a = u = i;
                        b = v = j;
                    }
                }
            }
            while (parent[u] != 0)
                u = parent[u];
            while (parent[v] != 0)
                v = parent[v];
            if (u != v) {
                System.out.println(ne++ + "edge (" + a + "," + b + ") =" + min);
                mincost += min;
                parent[v] = u;
            }
            cost[a][b] = cost[b][a] = 999;
        }
        System.out.println("Minimum cost :" + mincost);
    }
}
```