```
import java.util.Scanner;
/*
        Time complexity: O(E * log(V))
        Space compleity: O(V^2)

        where E is the number of edges in the graph and
        V is the number of vertices in the graph
 */

public class Solution {

        private static void dijkstra(int[][] adjacencyMatrix) {
                int v = adjacencyMatrix.length;
                boolean visited[] = new boolean[v];
                int distance[] = new int[v];
                distance[0] = 0;
                for (int i = 1; i < v; i++) {
                        distance[i] = Integer.MAX_VALUE;
                }

                for (int i = 0; i < v - 1; i++) {
                        // Find Vertex with Min distance
                        int minVertex = findMinVertex(distance, visited);
                        visited[minVertex] = true;
                        // Explore neighbors
                        for (int j = 0; j < v; j++) {
                                if (adjacencyMatrix[minVertex][j] != 0 && !visited[j] && distance[minVertex] != Integer.MAX_VALUE)
{
                                        int newDist = distance[minVertex] + adjacencyMatrix[minVertex][j];
                                        if (newDist < distance[j]) {
                                                distance[j] = newDist;
                                        }
                                }
                        }
                }

                // Print
                for (int i = 0; i < v; i++) {
                        System.out.println(i + " " + distance[i]);
                }

        }

        private static int findMinVertex(int[] distance, boolean visited[]) {

                int minVertex = -1;
                for (int i = 0; i < distance.length; i++) {
                        if (!visited[i] && (minVertex == -1 || distance[i] < distance[minVertex])) {
```

```
                        minVertex = i;
                }
        }
        return minVertex;
    }

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);
        int v = s.nextInt();
        int e = s.nextInt();
        int adjacencyMatrix[][] = new int[v][v];
        for (int i = 0; i < e; i++) {
            int v1 = s.nextInt();
            int v2 = s.nextInt();
            int weight = s.nextInt();
            adjacencyMatrix[v1][v2] = weight;
            adjacencyMatrix[v2][v1] = weight;
        }
        dijkstra(adjacencyMatrix);
    }

}
```