```
import java.util.Arrays;

public class Main {
    private static final int V = 5;

    int minKey(int key[], Boolean mstSet[]) {
        int min = Integer.MAX_VALUE, min_index = -1;

        for (int v = 0; v < V; v++)
            if (!mstSet[v] && key[v] < min) {
                min = key[v];
                min_index = v;
            }

        return min_index;
    }

    void printMST(int parent[], int graph[][]) {
        System.out.println("Edge \tWeight");
        for (int i = 1; i < V; i++)
            System.out.println(parent[i] + " - " + i + "\t" + graph[i][parent[i]]);
    }

    void primMST(int graph[][]) {
        int parent[] = new int[V];
        int key[] = new int[V];
        Boolean mstSet[] = new Boolean[V];

        for (int i = 0; i < V; i++) {
            key[i] = Integer.MAX_VALUE;
            mstSet[i] = false;
        }

        key[0] = 0;
        parent[0] = -1;

        for (int count = 0; count < V - 1; count++) {
            int u = minKey(key, mstSet);
            mstSet[u] = true;

            for (int v = 0; v < V; v++)
                if (graph[u][v] != 0 && !mstSet[v] && graph[u][v] < key[v]) {
                    parent[v] = u;
                    key[v] = graph[u][v];
                }
        }

        printMST(parent, graph);
    }

    public static void main(String[] args) {
        Main t = new Main();
        int graph[][] = new int[][]{{0, 2, 0, 6, 0},
                                    {2, 0, 3, 8, 5},
                                    {0, 3, 0, 0, 7},
                                    {6, 8, 0, 0, 9},
                                    {0, 5, 7, 9, 0}};

        t.primMST(graph);
    }
}
```

Output

```
Edge    Weight
0 - 1   2
1 - 2   3
0 - 3   6
1 - 4   5
```