

TARUN
19BCE7578
CSE3004

source History Connection:

```
1  import java.util.*;
2  import java.lang.*;
3  import java.io.*;
4  class Prims {
5  private static final int V = 5;
6  int minKey(int key[], Boolean mstSet[])
7  {
8  int min = Integer.MAX_VALUE, min_index = -1;
9  for (int v = 0; v < V; v++)
0  if (mstSet[v] == false && key[v] < min) {
1  min = key[v];
2  min_index = v;
3  }
4  return min_index;
5  }
6  void printMST(int parent[], int graph[][])
7  {
8  System.out.println("Edge \tWeight");
9  for (int i = 1; i < V; i++)
0  System.out.println(parent[i] + " - " + i + "\t" + graph[i][parent[i]]);
1  }
2  void primMST(int graph[][])
3  {
4  int parent[] = new int[V];
5  int key[] = new int[V];
6  Boolean mstSet[] = new Boolean[V];
7  for (int i = 0; i < V; i++) {
8  key[i] = Integer.MAX_VALUE;
9  mstSet[i] = false;
0  }
1  key[0] = 0;
2  parent[0] = -1;
3  for (int count = 0; count < V - 1; count++) {
4  int u = minKey(key, mstSet);
5  mstSet[u] = true;
6  for (int v = 0; v < V; v++)
7  if (graph[u][v] != 0 && mstSet[v] == false && graph[u][v] < key[v]) {
8  parent[v] = u;
9  key[v] = graph[u][v];
0  }
1  }
2  printMST(parent, graph);
3  }
4  public static void main(String[] args)
5  {
6  }
```

```
Start Page x project.java x newSQLTemplate.sql x
Source History Connection:
25 int key[] = new int[V];
26 Boolean mstSet[] = new Boolean[V];
27 for (int i = 0; i < V; i++) {
28     key[i] = Integer.MAX_VALUE;
29     mstSet[i] = false;
30 }
31 key[0] = 0;
32 parent[0] = -1;
33 for (int count = 0; count < V - 1; count++) {
34     int u = minKey(key, mstSet);
35     mstSet[u] = true;
36     for (int v = 0; v < V; v++)
37         if (graph[u][v] != 0 && mstSet[v] == false && graph[u][v] < key[v]) {
38             parent[v] = u;
39             key[v] = graph[u][v];
40         }
41     }
42     printMST(parent, graph);
43 }
44 public static void main(String[] args)
45 {
46     Prims t = new Prims();
47     int graph[][] = new int[][] { { 0, 2, 0, 6, 0 },
48     { 2, 0, 3, 8, 5 },
49     { 0, 3, 0, 0, 7 },
50     { 6, 8, 0, 0, 9 },
51     { 0, 5, 7, 9, 0 } };
52     t.primMST(graph);
53 }
54 }
```

Output - JavaApplication15 (run) x JavaApplication22.java x DigitalCamera.java x JavaApplication25.java

```
run:
please enter customer name
tarun
please enter membership type
gold
enter the price that you have purchased
200000
the discount is :40000
the price after the discount is 160000
-----visit again-----
BUILD SUCCESSFUL (total time: 23 seconds)
```

Time complexity

Worst case

Using binary heap $\rightarrow O(E \log v)$

Using Fibonacci heap $\rightarrow O(E + v \log v)$

Overall time complexity

$\Rightarrow O(E=v) \times O(\log v)$

$\Rightarrow O((E+v) \log v)$

$\Rightarrow O(E \log v)$