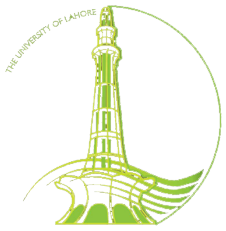

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

Data Structure and Algorithms

Lab Report

Name:	Asad Abbas
Registration #:	CSU-F16-109
Lab Report #:	11
Dated:	28-06-2018
Submitted To:	Mr. Usman Ahmed

The University of Lahore, Islamabad Campus
Department of Computer Science & Information Technology

Experiment # 11
Prims Minimum Spanning Tree

Objective
Understand the concept of prim Minimum spanning trees.

Software Tool
1. Dev C++

1 Theory

We have discussed Kruskals algorithm for Minimum Spanning Tree. Like Kruskals algorithm, Prims algorithm is also a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets, and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST
. A group of edges that connects two set of vertices in a graph is called cut in graph theory. So, at every step of Prims algorithm, we find a cut (of two sets, one contains the vertices already included in MST and other contains rest of the verices), pick the minimum weight edge from the cut and include this vertex to MST Set (the set that contains already included vertices).

Algorithm :
1) Create a set mstSet that keeps track of vertices already included in MST.
2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.
3) While mstSet doesnt include all vertices
.a) Pick a vertex u which is not there in mstSet and has minimum key value.
.b) Include u to mstSet.
.c) Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if weight of edge u-v is less than the previous key value of v, update the key value as weight of u-v

The idea of using key values is to pick the minimum weight edge from cut. The key values are used only for vertices which are not yet included in MST, the key value for these vertices indicate the minimum weight edges connecting them to the set of vertices included in MST.

2 Program

```
#include <stdio.h>
#include <limits.h>

#define V 5
int minKey(int key[], bool mstSet[])
{
   int min = INT_MAX, min_index;

   for (int v = 0; v < V; v++)
     if (mstSet[v] == false && key[v] < min)
         min = key[v], min_index = v;

   return min_index;
}

int printMST(int parent[], int n, int graph[V][V])
{
   printf("Edge   Weight\n");
   for (int i = 1; i < V; i++)
      printf("%d - %d    %d \n", parent[i], i, graph[i][parent[i]]);
}
void primMST(int graph[V][V])
{
     int parent[V];
     int key[V];
     bool mstSet[V];

     for (int i = 0; i < V; i++)
```

```
        key[i] = INT_MAX, mstSet[i] = false;

    key[0] = 0;     // Make key 0 so that this vertex is picked as first
    parent[0] = -1; // First node is always root of MST

    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] && mstSet[v] == false && graph[u][v] <
key[v])
              parent[v]  = u, key[v] = graph[u][v];
    }
    printMST(parent, V, graph);
}

int main()
{

   int graph[V][V] = {{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},
                     };
    primMST(graph);

    return 0;

}
```