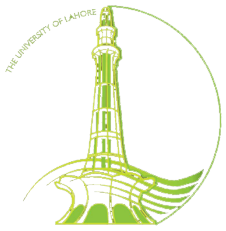

THE UNIVERSITY OF LAHORE
ISLAMABAD CAMPUS

DATA STRUCTURE
(CS13217)

Lab Report

Name:	Zain ul abideen
Registration #:	SEU-F16-133
Lab Report #:	12
Dated:	28-06-2018
Submitted To:	Mr. Usman Ahmed

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

Experiment # 1
Implementing minimum spanning tree algorothms.

Objective
To understand and implement the minimum spanning tree algorithms.

Software Tool
1.
dev c++

1 Theory

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted (un)directed graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. That is, it is a spanning tree whose sum of edge weights is as small as possible. More generally, any edge-weighted undirected graph (not necessarily connected) has a minimum spanning forest, which is a union of the minimum spanning trees for its connected components.

There are quite a few use cases for minimum spanning trees. One example would be a telecommunications company trying to lay cable in a new neighborhood. If it is constrained to bury the cable only along certain paths (e.g. roads), then there would be a graph containing the points (e.g. houses) connected by those paths. Some of the paths might be more expensive, because they are longer, or require the cable to be buried deeper; these paths would be represented by edges with larger weights. Currency is an acceptable unit for edge weight there is no requirement for edge lengths to obey normal rules of geometry such as the triangle inequality. A spanning tree for that graph would be a subset of those paths that has no cycles but still connects every house; there might be several spanning trees possible. A minimum spanning tree would be one with the lowest total cost, representing the least expensive path for laying the cable.

.

2 Task

2.1 procedure: Task 1

```
#include <stdio.h>
#include <limits.h>
 using namespace std;

#define V 8

int minKey(int key[], bool mstSet[])
{
   // Initialize min value
   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;
    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] && mstSet[v] == false && graph[u][v] <
key[v])
            parent[v]  = u, key[v] = graph[u][v];
    }

    // print the constructed MST
    printMST(parent, V, graph);
}



int main()
{

   int graph[V][V] = {{1, 8, 0, 0, 0,10,0,5},
                      {8, 0, 4, 0, 4,4,0,4},
                      {0, 4, 0, 3, 0,3,0,0},
                      {0, 0, 3, 0, 1,6,2,0},
                      {0, 4, 0, 1, 0,0,3,0},
                         {10, 4, 3, 6, 0,0,0,0},
                      {0, 0, 0, 2, 3,0,0,3},
```

```
                                {5, 4, 0, 0, 0,0,3,0},


                                };

    primMST(graph);

    return 0;
}
```

3 Conclusion

In this lab we perform the minimum spanning tree algorithms i understand the MST the sum of the weighted of its edge of the team nodes.