



THE UNIVERSITY
OF LAHORE
**ISLAMABAD
CAMPUS**

Data Structure And Algorithm

Lab Report

Name: Ch Hasnain Zafar
Registration #: SEU-F16-104
Lab Report #: 12
Dated: 25-06-2018
Submitted To: Mr. Usman Ahmed

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

Experiment # 1

Minimum Spanning Tree Algorithm

Objective

To understand and implement the Minimum Spanning Tree Algorithm Problem.

Software Tool

1. DEV C++

1 Theory

Given an undirected and connected graph , a spanning tree of the graph is a tree that spans (that is, it includes every vertex of) and is a subgraph of (every edge in the tree belongs to)

The cost of the spanning tree is the sum of the weights of all the edges in the tree. There can be many spanning trees. Minimum spanning tree is the spanning tree where the cost is minimum among all the spanning trees. There also can be many minimum spanning trees.

Minimum spanning tree has direct application in the design of networks. It is used in algorithms approximating the travelling salesman problem, multi-terminal minimum cut problem and minimum-cost weighted perfect matching. Other practical applications are:

- a- Cluster Analysis
- b- Handwriting recognition
- c- Image segmentation

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[])
{
    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\t\tWeight\n");
    for (int i = 1; i < V; i++)
        printf("%d\t-%d\t\t\t%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];
    }
    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;
    }

```