Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment – 5

Name: Hemil Sanjaybhai Panchiwala
Enrollment No.: 18114031
Branch: Computer Science & Engineering
Sub Batch: O2

Problem Statement 1

Write a C++ program to perform addition and multiplication of two polynomial expressions using any data structure chosen from STL. The polynomial expressions are of the form ax2 + bx + c, where a, b and c are real constants. The inputs for 2x2 + 5x + 6 and 2x3 + 5x2 + 1x + 1 are shown below (real constants followed by their power of x).

Data Structures used :-

Map, Array

Algorithm

- **I have created a hashmap for two polynomials with the value at a particular degree would be its coefficient i.e. hash[degree] = coefficient.**

- **Then, for addition, I have taken the maximum degree of two and added the same index from hashmap.**

- **For multiplication, I have applied the simple multiplication which we follow in daily life using indexes of hashmap as degrees.**

Snapshots

```
Enter number of terms in first polynomial: 3
Enter the Coefficient and Power:
2 2
5 1
6 0
Enter number of terms in second polynomial: 4
Enter the Coefficient and Power:
2 3
5 2
1 1
1 0
Enter 1 to add:
Enter 2 to multiply:
Enter 3 to exit:
1
After addition:
Coefficients        Power
2                   3
7                   2
6                   1
7                   0
Enter 1 to add:
Enter 2 to multiply:
Enter 3 to exit:
2
After multiplication:
Coefficients        Power
4                   5
20                   4
39                   3
37                   2
11                   1
6                   0
```

Time taken:- 0.001041s

Problem Statement 2

Given a set of nodes connected to each other in the form of a weighted undirected graph G, find the minimum spanning tree (MST). A spanning tree T of an undirected graph G is a subgraph that is a tree which includes all of the vertices of G, with minimum possible number of edges. G may have more than one spanning trees. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree. A minimum spanning tree (MST) is a spanning tree whose weight is less than or equal to that of every other spanning tree.

For given input graph (given as a CSV file having the format as shown in the example below), implement Kruskal's algorithm in C++ program using UNION FIND data structures (without using STL) and show all the edges of the MST as output in both the command line and in the "dot file", where DOT is a graph description language. Also, print the total edge weight of the MST. For more details follow this link https://www.graphviz.org/doc/info/lang.html. Further use the "dot file" file to visualize the output graph in .pdf or .png file using Graphviz.

Data Structures used :-

Union set, Vectors

Algorithm

- **Sort all the edges from low weight to high weight.**

- **Take the edge with the lowest weight and add it to the spanning tree. If adding the edge created a cycle, then reject this edge.**

- **Keep adding edges until we reach all vertices.**

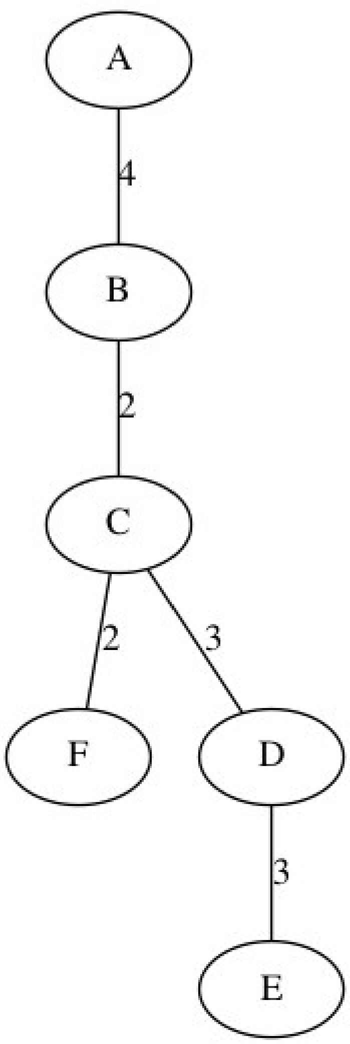
- **Here, I have used union-find algorithm to check if adding any edge create a loop or not. This algorithm divides the vertices into clusters and allows us to check if two vertices belong to the same cluster or not and hence decide whether adding an edge creates a cycle.**

Time taken:- 0.001137s

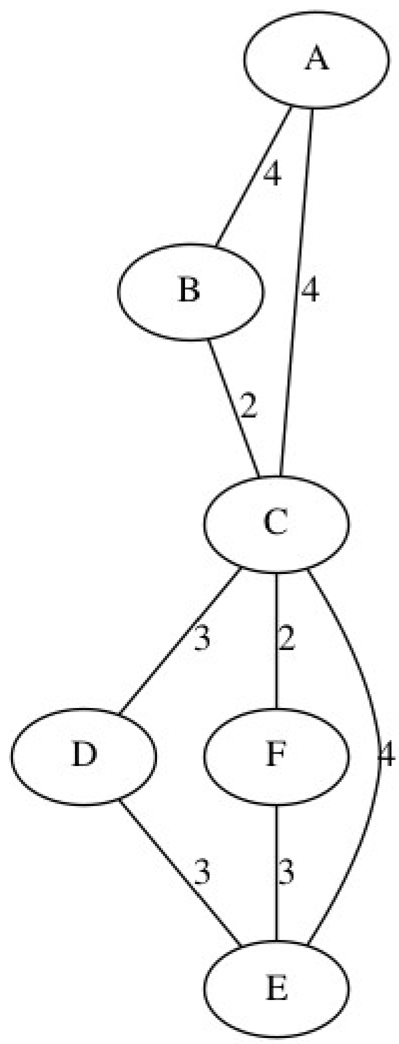
Snapshots

```
Node1 Node2 Weight
A B 4
A C 4
B C 2
C D 3
C F 2
C E 4
D E 3
F E 3
Node1 Node2 Weight
B C 2
C F 2
C D 3
D E 3
A B 4
```

<u>Output MST Tree</u>



<u>Input Tree</u>



Problem Statement 3

Write a C++ program to implement Prim's algorithm for a given input graph (given as a CSV file having the format as shown in the example below) using Fibonacci heap data structure to find the minimum spanning tree (MST). You can use STL for the data structure used in this C++ program.

It is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm generates the MST by adding one vertex at a time, starting from an arbitrary vertex. At each step the cheapest possible edge weight is chosen from the already selected vertex. These algorithms find the minimum spanning forest in a possibly disconnected graph; in contrast, the most basic form of Prim's algorithm only finds minimum spanning tree in connected graphs.

Show all the edges of the MST as the output in command line. Also, print the total edge weight of the MST. Use Newick file format (https://en.wikipedia.org/wiki/Newick_format) for visualization of the MST in ETE Toolkit (http://etetoolkit.org/).

Data Structures used :-

Fibonacci heap

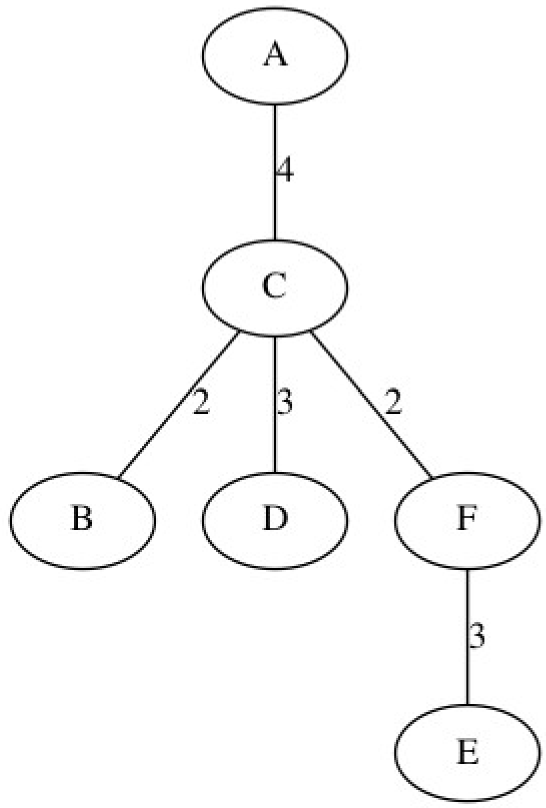
Algorithm

- Randomly choose any vertex.
- We usually select and start with a vertex that connects to the edge having least weight.
- Find all the edges that connect the tree to new vertices, then find the least weight edge among those edges and include it in the existing tree.
- If including that edge creates a cycle, then reject that edge and look for the next least weight edge.
- Keep repeating last two steps until all the vertices are included and Minimum Spanning Tree (MST) is obtained.\

Snapshots

Prims MST Tree

```
Node1 Node2 Weight
A B 4
A C 4
B C 2
C D 3
C F 2
C E 4
D E 3
F E 3
Node1 Node2 Weight
B C 2
A C 4
C D 3
E F 3
C F 2
```



Time taken:- 0.000664s