Lab Assignment - 5

Name: Shubhang Tripathi
Enrollment No: 18114074
Batch: O3

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 ax 2 + bx + c, where a, b and c are real constants.

DATA STRUCTURES USED:

- 2-D dynamic arrays

ALGORITHMS USED:

- Used addition of partial products for Multiplication

```
$ ./q1
Enter first Polynomial
No. of terms in the expression :3
2 2
5 1
6 0
Enter second Polynomial
No. of terms in the expression :4
2 3
5 2
1 1
1 0
Enter 1 to add or 2 for multiply
1
2       3
7       2
6       1
7       0
Enter 1 to add or 2 for multiply
2
4       5
20      4
39      3
37      2
11      1
6       0
Enter 1 to add or 2 for multiply
```

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. Further use the "dot file" file to visualize the output graph in .pdf or .png file using Graphviz.

DATA STRUCTURES USED:

- Arrays
- Singly-Linked List
- Graphs
- Union- Find data structures

ALGORITHMS USED:

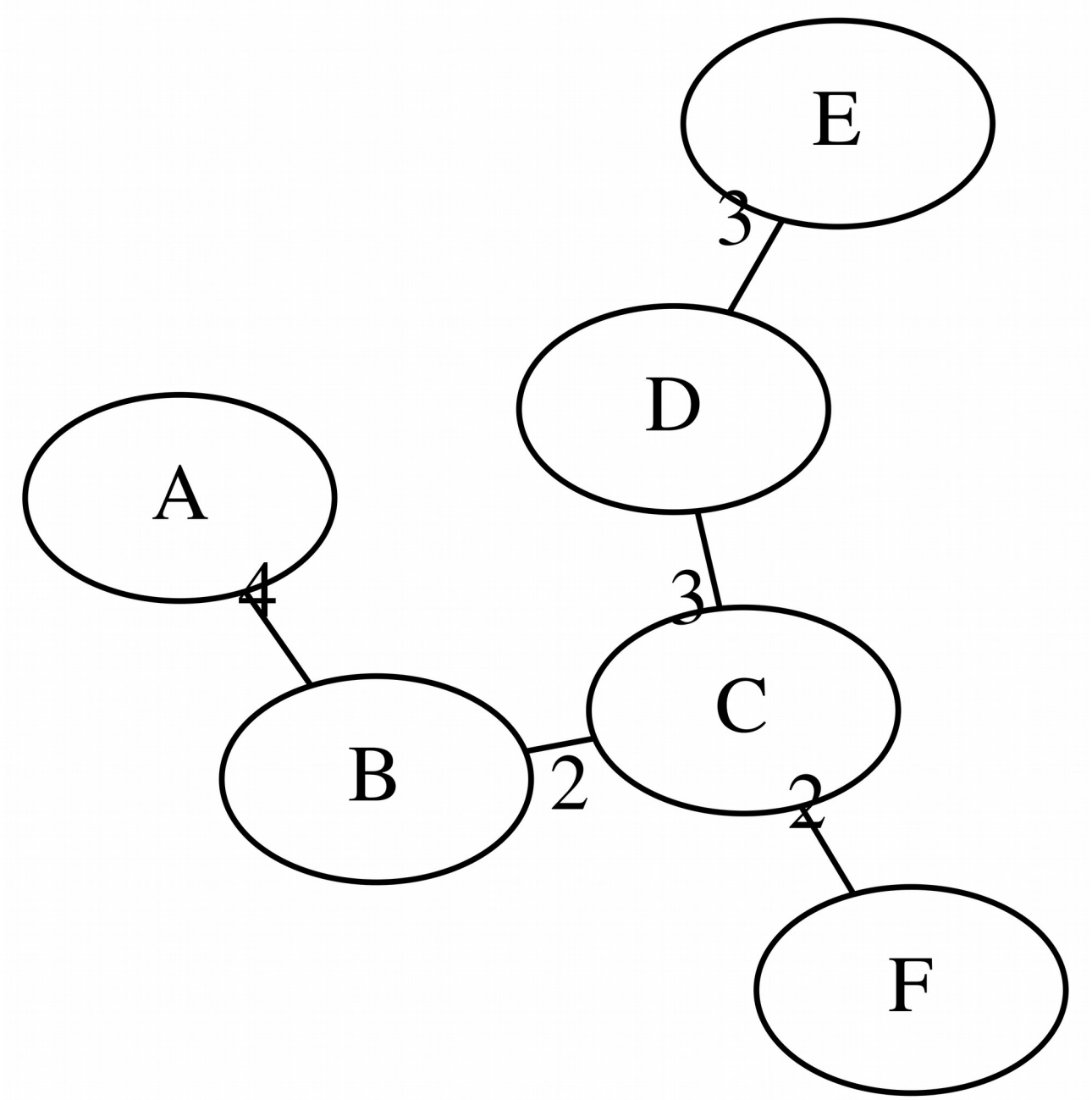
- Krushkal's Algorithm
- Union and find algorithms to implement Krushkal's algorithm

```
$ ./q2

B,C,2
C,F,2
C,D,3
D,E,3
A,B,4

The total edge weight of the MST is : 14
thefox@thebunker:~/Desktop/CSN261_Assign/csn261_assign5
$ ls q2.png
q2.png
thefox@thebunker:~/Desktop/CSN261_Assign/csn261_assign5
$ xdg-open q2.png
```

Output of Graphviz for the Minimum Spanning Tree



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. 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 for visualization of the MST in ETE Toolkit

DATA STRUCTURES USED:

- 2-D arrays
- Vectors
- Graphs
- **Fibonacci Heap**
- Priority Queue

ALGORITHMS USED:

- Prim's Algorithm
- Use of Fibonacci heap to create the required min-heap
- Use of vectors to get newwick fornmat output
- Use of adjacency matrix to store the graph
- Recursive algorithm to get the newwick format using DFS algorithm

```
$ ./q3
A,B,4
B,C,2
C,D,3
C,F,2
E,F,3

The total edge weight of the MST is : 14
thefox@thebunker:~/Desktop/CSN261_Assign/csn261_assign5
$ cat prim.nw
(((D:3,(E:3)F:2)C:2)B:4)A;
thefox@thebunker:~/Desktop/CSN261_Assign/csn261_assign5
$ python etePrim.py

         /-D
-A /B /C|
         \F /-E
```