

# L5 Report

Name: Priyanshu Garg

Enrollment No.: 18114058

Branch: Computer Science and Engineering

Batch: O3

### Problem1:

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 structure:

In this question, program used map data structure as in map we can store how many times a particular power comes that's why it is easy to keep track of power of  $x$  and their coefficients.

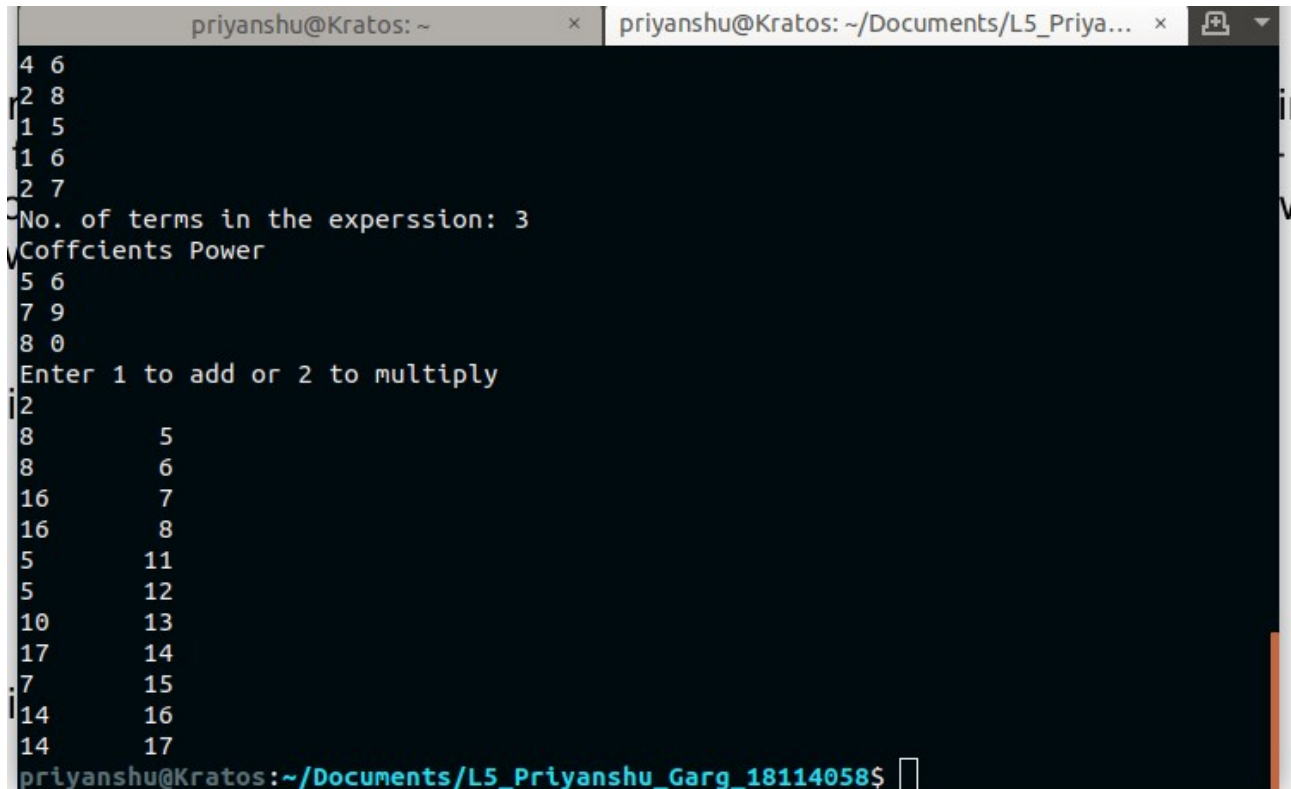
### Algorithm:

Firstly, program makes 5 maps (ordered), to store the values of poly1, poly2, poly\_add, poly\_mul. Then it takes input from the user to store those values in the respective maps, after this user inputs whether to add or multiply.

For add, program simply stores the values in map with key as power of  $x$ , for which they are coefficient. Then simply runs a for loop to add common key elements and add the extra ones in poly\_add map.

For multiply, program used brute force method to multiply each term with the other terms and store in poly\_temp and update in the poly\_mul after completing one cycle.

### Screenshots:



```
priyanshu@Kratos: ~  
4 6  
2 8  
1 5  
1 6  
2 7  
No. of terms in the expression: 3  
Coefficients Power  
5 6  
7 9  
8 0  
Enter 1 to add or 2 to multiply  
2  
8 5  
8 6  
16 7  
16 8  
5 11  
5 12  
10 13  
17 14  
7 15  
14 16  
14 17  
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$
```

```

priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ g++ problem1.cpp
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ ./a.out
No. of terms in the experssion: 5
Coefficients Power
4 6
2 8
1 5
1 6
1 7
No. of terms in the experssion: 3
Coefficients Power
1 6
1 9
2 0
Enter 1 to add or 2 to multiply
1
2      0
1      5
2      6
1      7
2      8
1      9
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ 

```

#### Problem2:

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.

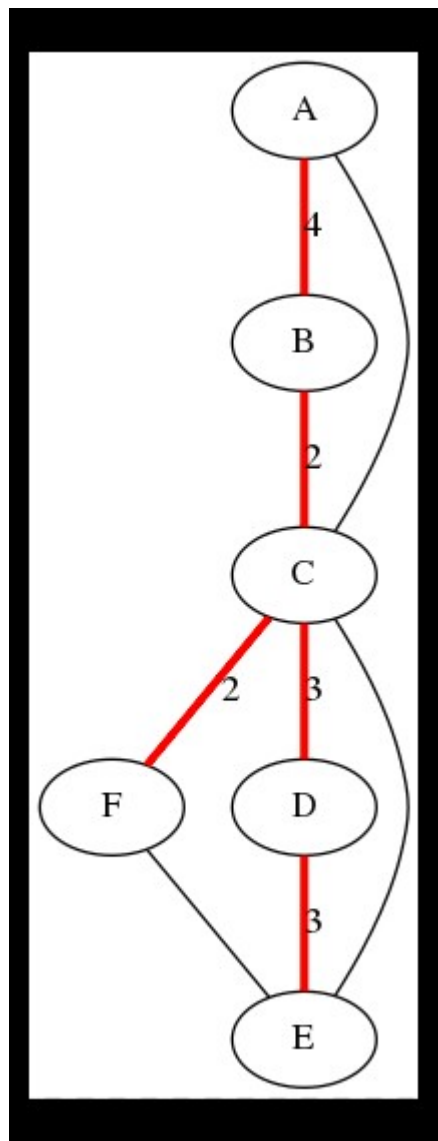
#### Data structures:

I used the concepts of graphs and UNION SET i such as making of root node tp fnd the union of two sets, also pairs and arrays are used to store the information about the given path.

#### Algorithm:

Firstly, program stores the nodes in pair and the weight of the path , after this sort the pairs according to their weights and starts transversing frommthe lowest weight, after this program used the concepts of union set to join disjoint set first such that all the noes are connected with each other with minimum MST.

#### Screenshots:



```
14
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ cat graph.dot
graph {
    B--C[label="2",weight="2",color=red,penwidth=3.0];
    C--F[label="2",weight="2",color=red,penwidth=3.0];
    C--D[label="3",weight="3",color=red,penwidth=3.0];
    D--E[label="3",weight="3",color=red,penwidth=3.0];
    F--E;
    A--B[label="4",weight="4",color=red,penwidth=3.0];
    A--C;
    C--E;
}
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$
```

```
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ ./a.out
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

Optimised answer:-
Node1 Node2 Weight
B C 2
C F 2
C D 3
D E 3
A B 4
14
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$
```

#### Problem3:

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.

#### Data structure:

Program used the heap and vector, pairs to store the values of edges and also extra queues are used to do calculations.

#### Algorithms:

I used the concepts of graphs and queues to execute Prim's algorithm

#### Screenshots:

```
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ g++ problem3.cpp
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$ ./a.out
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

Optimised answer:-
Node1 Node2 Weight
F C 2
C B 2
C D 3
D E 3
C A 4
14
priyanshu@Kratos:~/Documents/L5_Priyanshu_Garg_18114058$
```