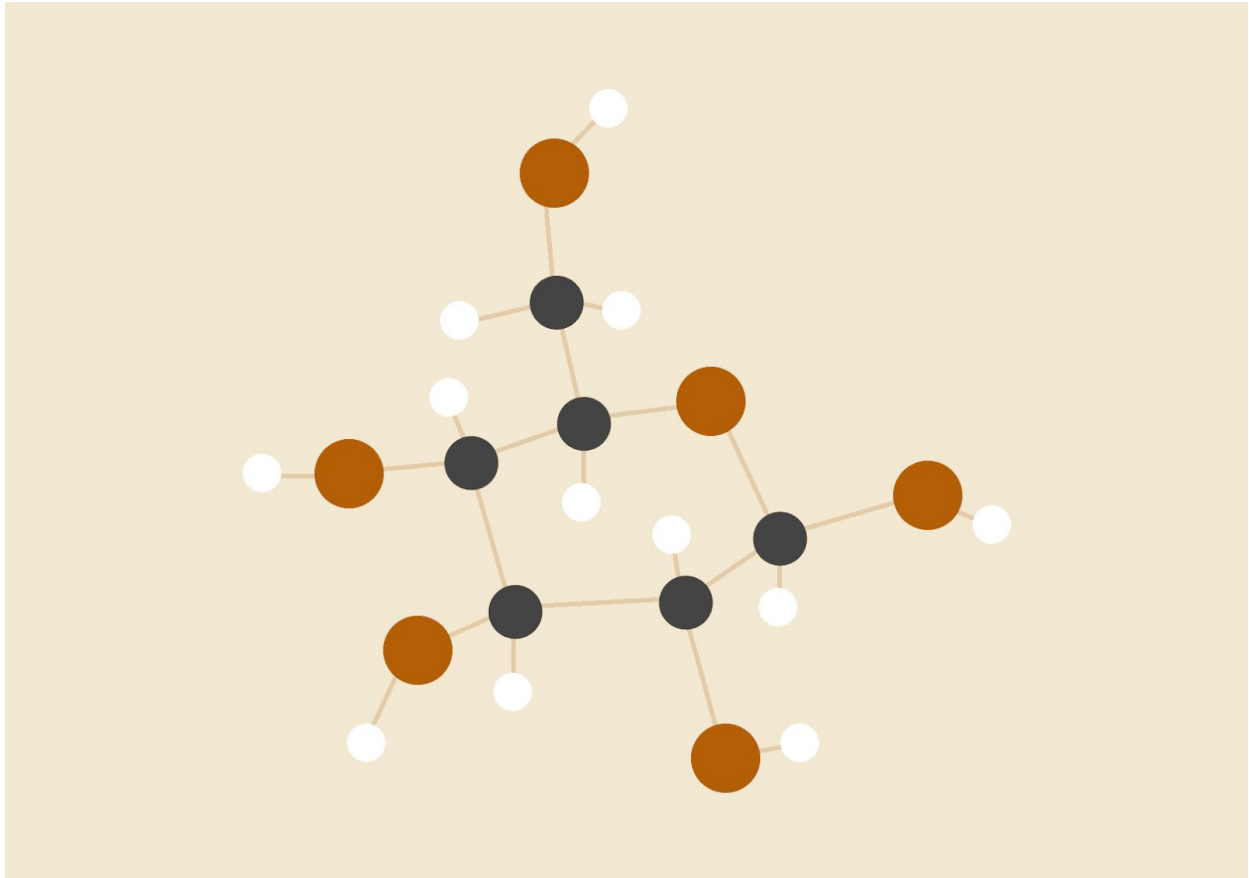


ASSIGNMENT 6

Data Structures Laboratory



Rishi Chordia

18118052

BTech CSE

Problem Statement 1:

Write a menu driven C++ program to implement a graph using adjacency list (linked list) without using STL.

Perform following operations on the graph. 1. Insert edge 2. BFS traversal 3. DFS traversal 4. Cycle finding in the graph 5. Calculate diameter of the graph

Algorithms and Implementation:

- **Floyd Warshall's algorithm used to find diameter**
- **Adjacency List made up of linked List was used to implement graph nodes**
- **Stacks Data Structures used for DFS**
- **Queue Data Structure used for BFS**
- **To find the diameter we have found the minimum distance of all pairs of node and then take the maximum of these using DFS**
- **Time has been calculate using the clock class in time.h**
- **GDB used for debugging**

Snapshots and Computation Time

```
Activities Terminal Oct 9 02:31
arki1418@rishi-G5: ~/CSN261/L6$ ./Q1

-----MENU-----

1. To insert an edge
2. BFS traversal
3. DFS traversal
4. To check if cycle present (connected graph)
5. Diameter of the graph
6. Exit

Your choice: 2
Enter the root node for BFS traversal:A

A B C I F D U

Time taken by program is : 0.000155 sec
```

```
-----MENU-----

1. To insert an edge
2. BFS traversal
3. DFS traversal
4. To check if cycle present (connected graph)
5. Diameter of the graph
6. Exit

Your choice: 3
Enter the root node for DFS traversal:A

A B I U D F C

Time taken by program is : 0.00014 sec
```

```
-----MENU-----  
  
1. To insert an edge  
2. BFS traversal  
3. DFS traversal  
4. To check if cycle present (connected graph)  
5. Diameter of the graph  
6. Exit  
  
Your choice: 4  
  
A B I U D F C  
  
True  
Time taken by program is : 0.00013 sec
```

```
-----MENU-----  
  
1. To insert an edge  
2. BFS traversal  
3. DFS traversal  
4. To check if cycle present (connected graph)  
5. Diameter of the graph  
6. Exit  
  
Your choice: 5  
  
The diameter is 3  
  
Time taken by program is : 0.00008 sec
```

Problem Statement 2:

A binomial heap is implemented as a set of binomial trees, which are defined recursively as follows: x A binomial tree of order 0 is a single node x A binomial tree of order k has a root node whose children are roots of binomial trees of orders $k-1, k-2, \dots, 2, 1, 0$ (in this order). x A binomial tree of order k has 2^k nodes, height k . Write a C++ program to implement a binomial heap using heap data structures (without using STL). Print the order of each binomial heap and use Graphviz to show the forest of binomial heap.

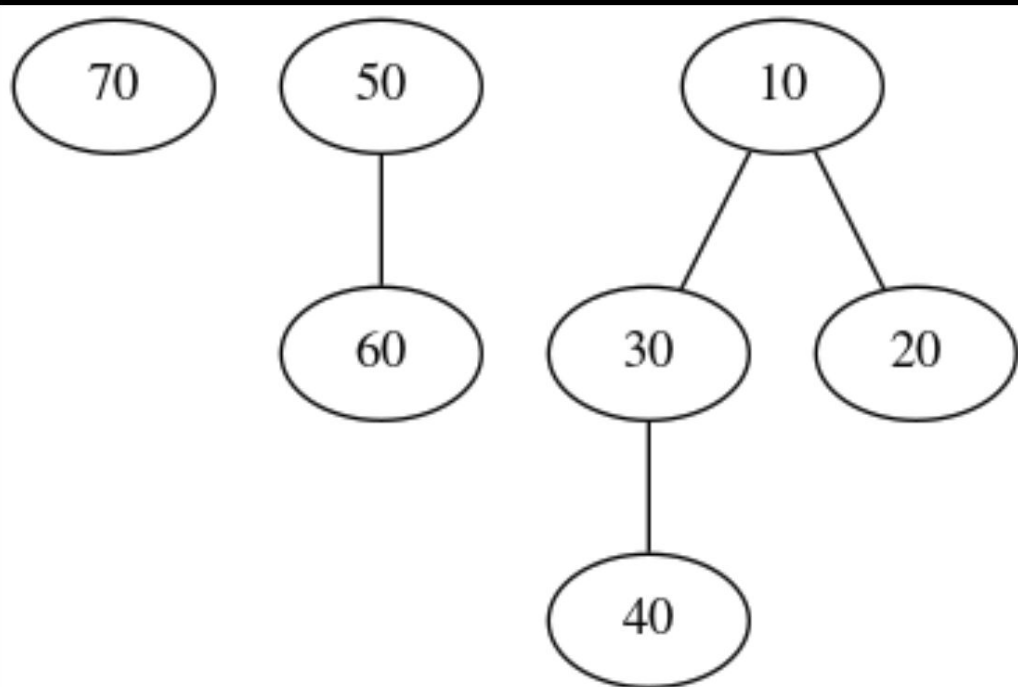
Algorithms and Implementation:

- Union Algorithms have been used to create binomial trees
- GraphViz used to show the graphs

Snapshots and Computation Time

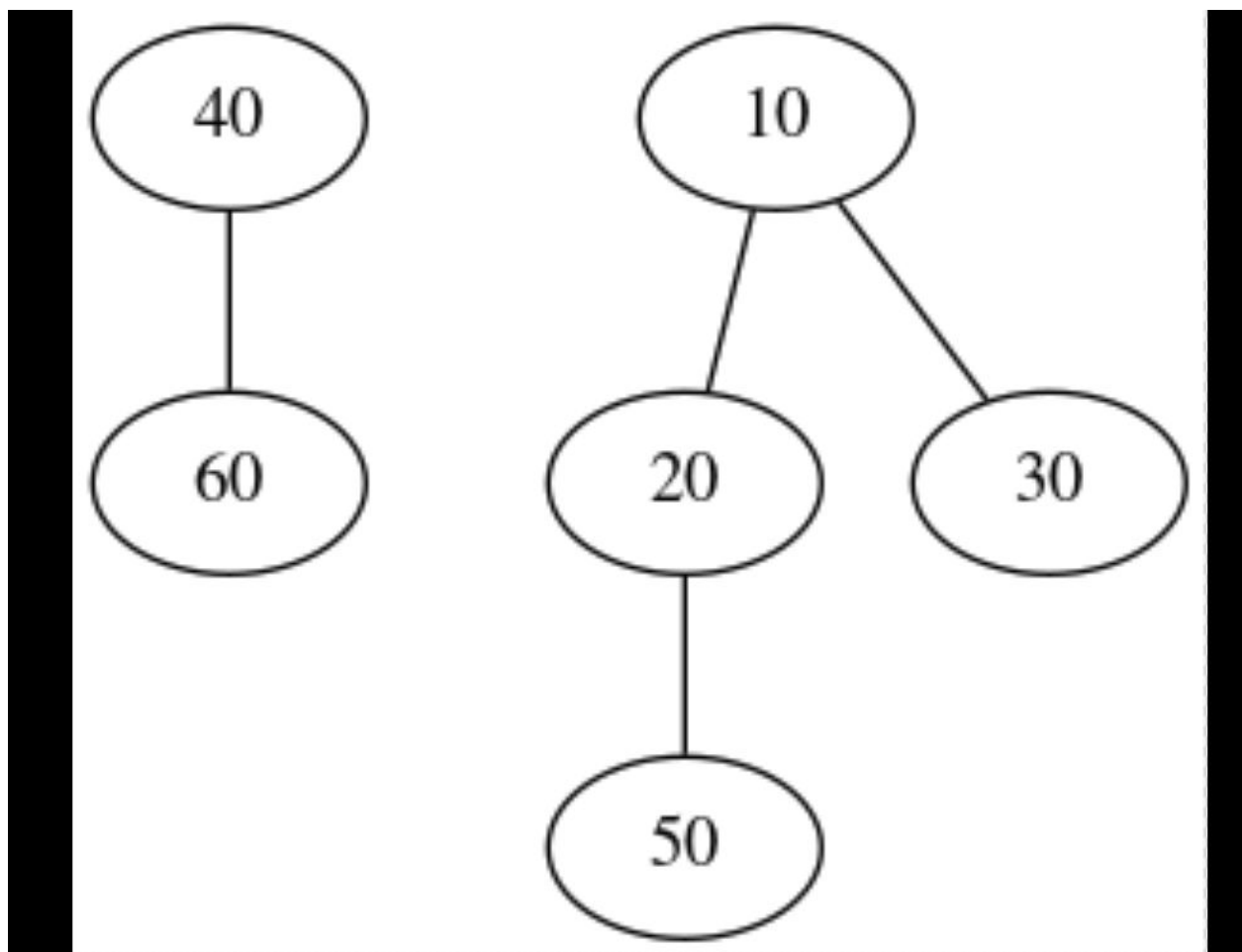
```
Activities Terminal Oct 9 02:10
arki1418@rishi-G5: ~/CSN261/L6$ ./Q2
7
10 20 30 40 50 60 70
Order: Elements
0 : 70
1 : 50 60
2 : 10 30 40 20

Time taken by program is : 0.000517 sec
arki1418@rishi-G5:~/CSN261/L6$
```



```
Activities Terminal ▾ Oct 9 02:11
arki1418@rishi-G5: ~/CSN261/L6
arki1418@rishi-G5:~/CSN261/L6$ ./Q2
6
10 30 20 50 40 60
Order: Elements
1 : 40 60
2 : 10 20 50 30

Time taken by program is : 0.000509 sec
arki1418@rishi-G5:~/CSN261/L6$
```



Problem Statement 3:

Write a C++ program to implement Bentley-Ottmann Algorithm to find and print all the intersection points of n given lines. Use of STL is allowed. The specific type of data structure that must be used include Priority Queue and BST. Using least square method find the linear fit of the M found intersection points and print the line in the form $ax+b$. The student should demonstrate this on a GUI using QT library. The input should be given in following format: 1. Input number of line segments, N 2. N lines where $2N$ points are provided, i.e., 2 points in each line

Algorithms and Implementation:

- Bentley-Ottman's Algorithm implemented using Line Sweep Technique
- Priority Queues along with Binary Search Trees have been used
- QT Libraries used to visualize the graph

Snapshots and Computation Time

```
Activities  Terminal  Oct 9 04:31
Enter the number of lines : 6
Enter the coordinates space separated
104 212 513 727
229 424 538 278
249 324 654 657
508 440 531 623
453 295 517 398
639 290 601 116
No. of intersection points: 4
(260.533,409.101)
(318.938,381.505)
(464.125,312.905)
(521.59,548.13)

```

number of intersections : 4

The fit line is of the form $0.293744x + 297.969$

