Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory
(Autumn 2019-2020)
Lab Assignment-6 (L6)

Name :- Harshit Verma **Enroll no:- 18114029**
Branch :- CSE **Sub Batch :- O2**

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.

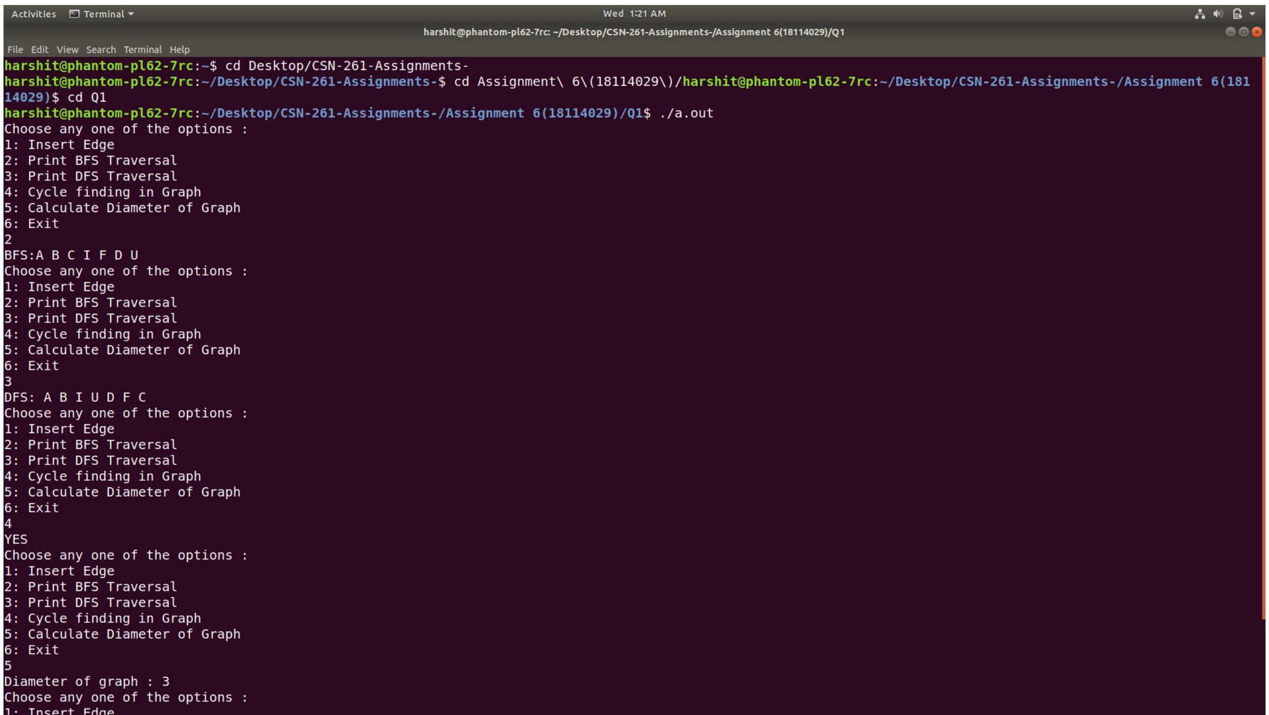
1. Insert edge
2. BFS traversal
3. DFS traversal
4. Cycle finding in the graph
5. Calculate diameter of the graph .

Data Structures used :

- Linked List
- Array
- Graph

Algortihm used :

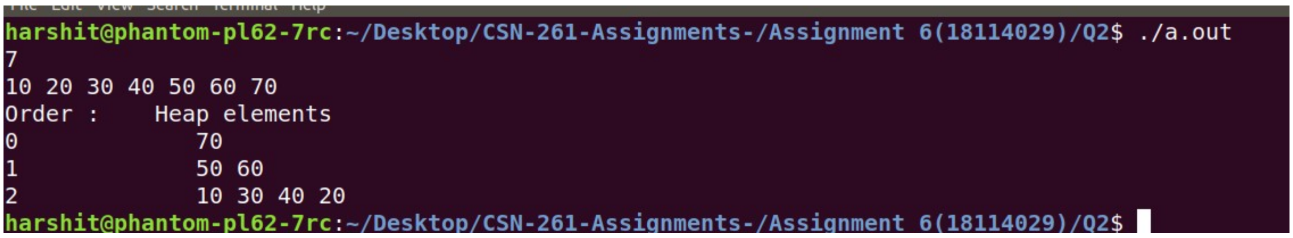
• We can use adjacency list using linked list to store edges .
• To do bfs , first maintain an array containing flag for visited node . Use queue to keep track of nodes , when you visit a node , push it onto the queue , then also visit all its neighbouring nodes and recursively apply this , also dont visit already visited nodes
• To do dfs , again maintain an array containing flag for visited node . Use stack to keep track of nodes .Print it. Then visit all of its child recursively . Dont visit already visited nodes .
• To find a cycle , apply dfs , if you visit a node that has been already visited before and its not immediate parent , then cycle is present
• To find diameter of graph apply BFS at every node , use this to find shortest path between nodes , now maximum of these shortest paths is the diameter .


Activities
Terminal
Wed 1:21 AM
harshit@phantom-pl62-7rc: ~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q1
File Edit View Search Terminal Help

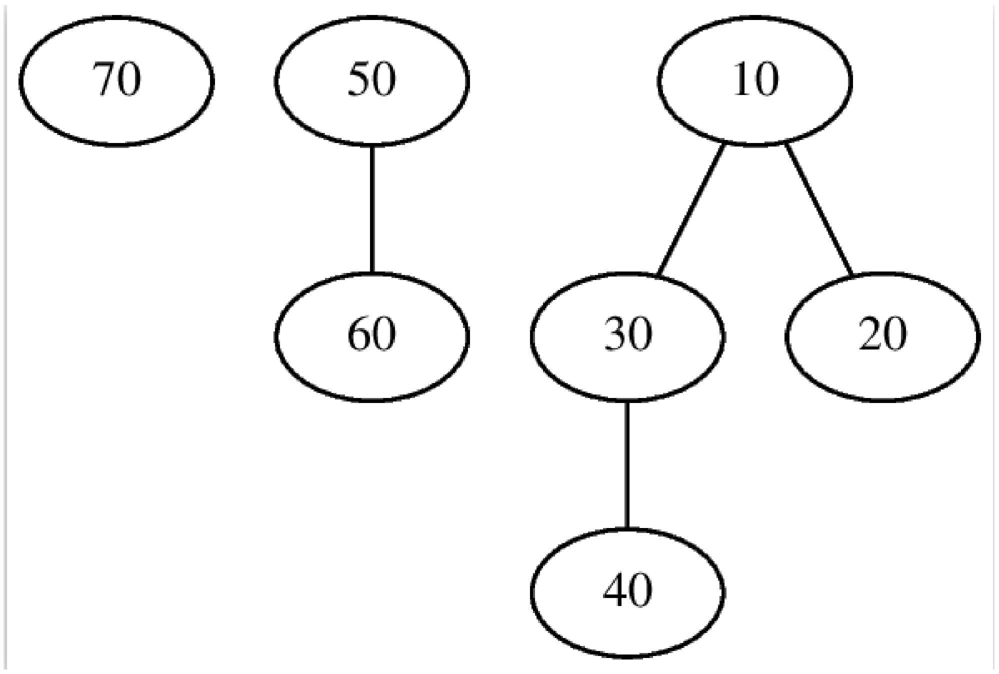
```
harshit@phantom-pl62-7rc:~$ cd Desktop/CSN-261-Assignments-
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-$ cd Assignment\ 6\(18114029\)/harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(181
14029)$ cd Q1
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q1$ ./a.out
Choose any one of the options :
1: Insert Edge
2: Print BFS Traversal
3: Print DFS Traversal
4: Cycle finding in Graph
5: Calculate Diameter of Graph
6: Exit
2
BFS:A B C I F D U
Choose any one of the options :
1: Insert Edge
2: Print BFS Traversal
3: Print DFS Traversal
4: Cycle finding in Graph
5: Calculate Diameter of Graph
6: Exit
3
DFS: A B I U D F C
Choose any one of the options :
1: Insert Edge
2: Print BFS Traversal
3: Print DFS Traversal
4: Cycle finding in Graph
5: Calculate Diameter of Graph
6: Exit
4
YES
Choose any one of the options :
1: Insert Edge
2: Print BFS Traversal
3: Print DFS Traversal
4: Cycle finding in Graph
5: Calculate Diameter of Graph
6: Exit
5
Diameter of graph : 3
Choose any one of the options :
1: Insert Edge
```

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, ..., 2, 1, 0 (in this order). x .A binomial tree of order k has 2k 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.

Data Structures:
• Binmoal heap
• Stack
Algorithm
• I have used stack to implement this .
• Its a collection of many binomial trees .
• Insert a new node as root .
• When two adjacent roots have same degree , make the one with the smaller key as the parent .
• Continue this



```
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q2$ ./a.out
7
10 20 30 40 50 60 70
Order :     Heap elements
0             70
1             50 60
2             10 30 40 20
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q2$
```

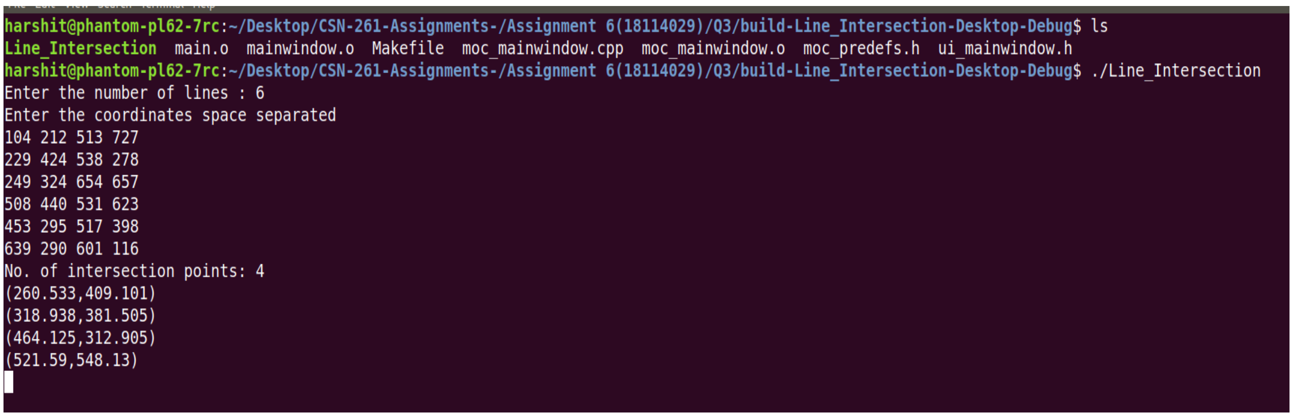


Write a C++ program to implement Bentley-Ottmann Algorithm to find and print all the intersection points of n given lines. Use of STL is allowed. The specific type of data structure that must be used include Priority Queue and BST. Using least square method find the linear fit of the M found intersection 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 .

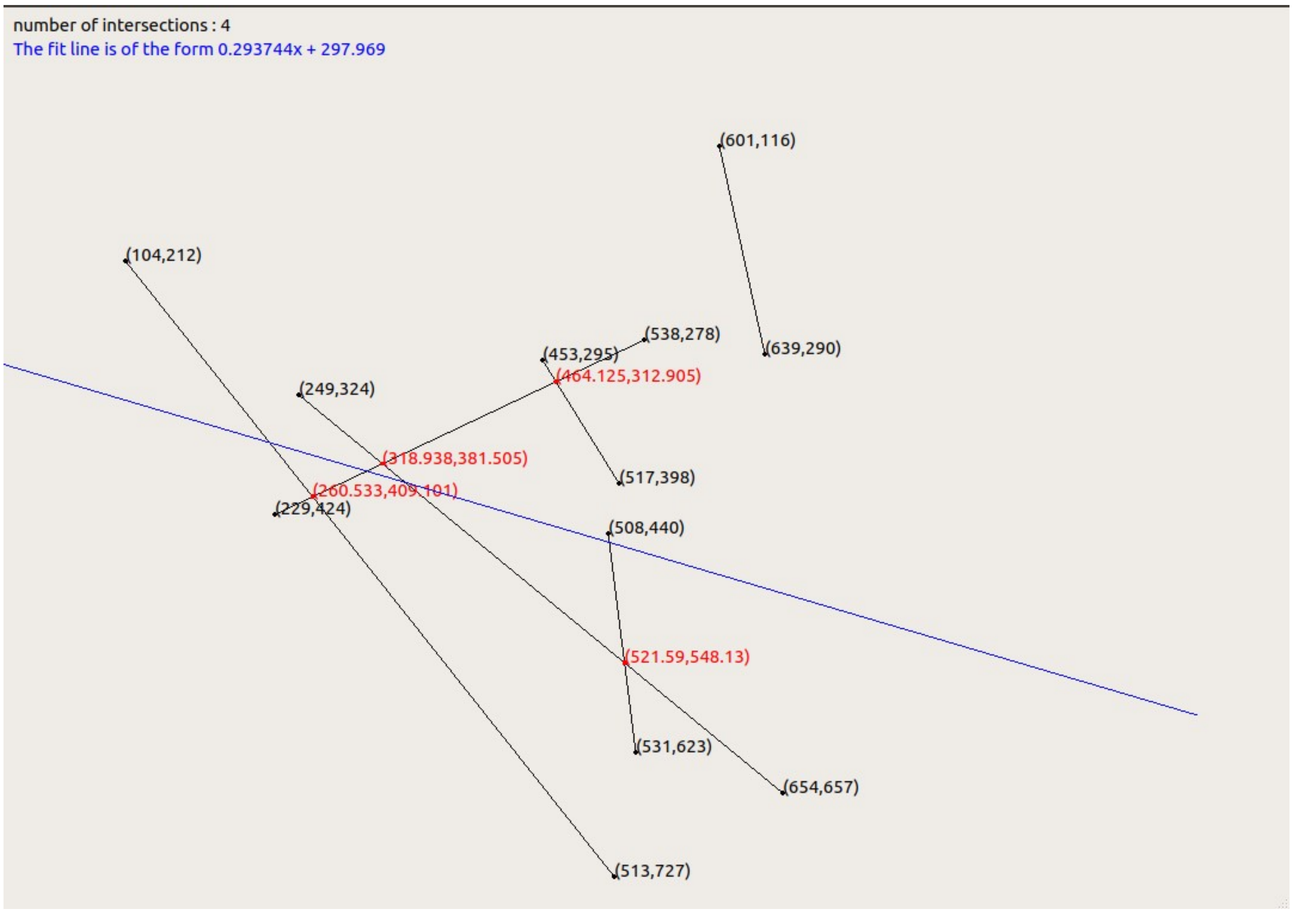
Data Structures used :
• Priority Queue
• BST
• Array
• Stack
• Heap

Algorithms:
• When we hit starting point of horizontal line segment, we insert the line (in our implementation, we will insert the starting point) in our set.
• When we hit end point of horizontal line segment, we remove the line segment (starting point of line segment in the implementation) from the set.
• When we hit a vertical line, we check for all the line segments in the set which lie between the starting and ending y coordinate of vertical line segment, i.e., if vertical line segment is indicated by (x1,y1) and (x1,y2), we check for horizontal line segments lying in the range (y1,y2).

Screenshots:



```
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q3/build-Line_Intersection-Desktop-Debug$ ls
Line_Intersection  main.o  mainwindow.o  Makefile  moc_mainwindow.cpp  moc_mainwindow.o  moc_predefs.h  ui_mainwindow.h
harshit@phantom-pl62-7rc:~/Desktop/CSN-261-Assignments-/Assignment 6(18114029)/Q3/build-Line_Intersection-Desktop-Debug$ ./Line_Intersection
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