Indian Institute of Technology Roorkee

Department of Computer Science and Engineering

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

Lab Assignment-6 (L6) **Date: September 25, 2019** **Duration: 2 Weeks**

General Instructions:

1. Every Lab Assignment will be performed by the students individually. No group formation is required and the evaluations will be done every week for the students individually.

Submission and Evaluation Instructions:

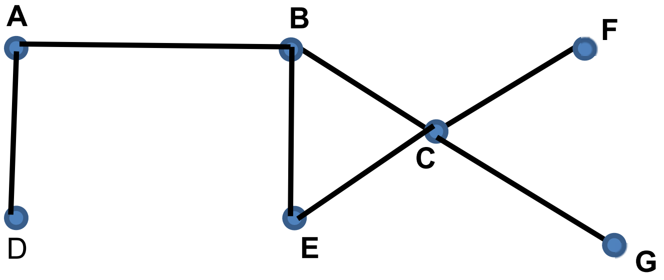
1. **Submit your** zipped folder (**<filename>.zip** or **<filename>.tar.gz**) through your account in Moodle through the submission link for this Lab Assignment in Moodle course site: https://moodle.iitr.ac.in/course/view.php?id=46.
2. **Hard deadline for Final submission in Moodle: October 9, 2019 (1:00 pm Indian Time).** For any submission after Final Deadline, 20% marks will be deducted (irrespective of it is delayed by a few seconds or a few days). The key to success is starting early. You can always take a break, if you finish early.
3. The submitted zipped folder (**<filename>.zip** or **<filename>.tar.gz**) must contain the following:
   (a) The source code files in a folder
   (b) A report file (**<filename>.DOC** or **<filename>.PDF**) should contain the details like:
      i. Title page with details of the student
      ii. Problem statements
      iii. Algorithms and data structures used in the implementation
      iv. Snapshots of running the codes for each problem
4. The submission by each student will be checked with others' submission to identify any copy case (using such detection software). If we detect that the code submitted by a student is a copy (partially or fully) of other's code, then the total marks obtained by one student will be divided by the total number of students sharing the same code.

Instructions for L6:

1. Objective of this Lab Assignment is to make the students familiar with different data structures while coding the programs in the C++ language to solve some real-life problems.
2. The students are expected to have a basic knowledge of data structures and the C++ programming language.
3. The student will have to demonstrate and explain the coding done for this Lab Assignment in the next laboratory class to be held on **October 9, 2019** for evaluation.

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

Input:



Output:

2. A B D E C F G
3. A B C G F E D
4. Yes
5. Diameter: 4

Problem Statement 2:

A binomial heap is implemented as a set of **binomial trees,** which are defined recursively as follows:

- A binomial tree of order 0 is a single node
- 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).
- A binomial tree of order *k* has 2^k nodes, height *k*.

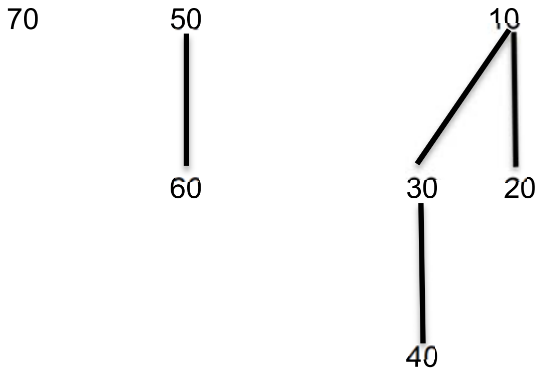
Write a C++ program to implement a binomial heap using heap data structures (without using STL). Print the order of each binomial heap and use Graphviz to show the forest of binomial heap.

Input
7
10, 20, 30, 40, 50, 60, 70

Output:
Order : Heap elements

```
0  :   70
1  :   50  60
2  :   10  30  40  20
```

3 binomial heap will be formed of order 0   1   and 2



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:

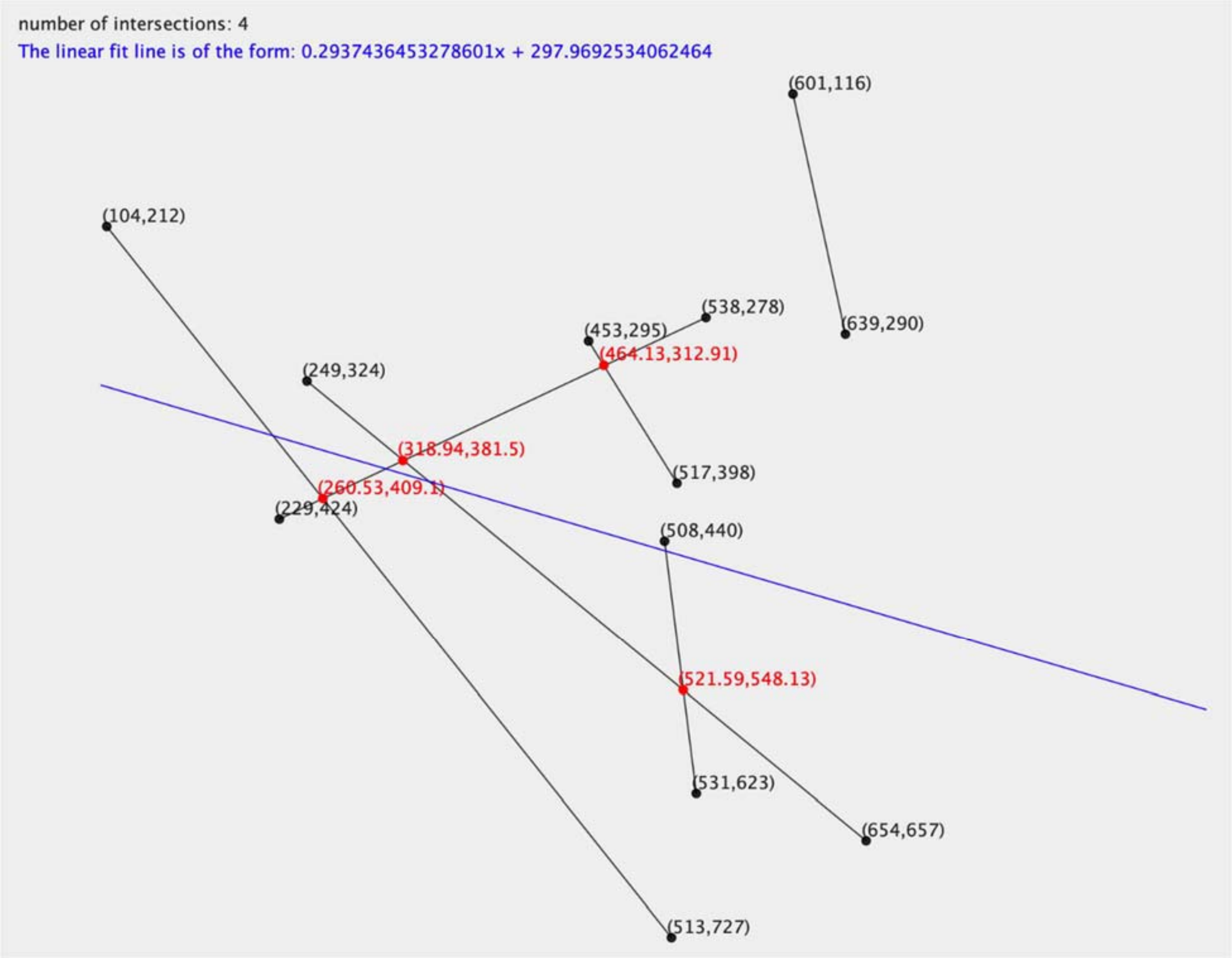
1. Input number of line segments, N
2. N lines where 2N points are provided, i.e., 2 points in each line

Sample Input:

```
N = 6
P1X P1Y P2X P2Y
104 212 513 727
229 424 538 278
249 324 654 657
508 440 531 623
453 295 517 398
639 290 601 116
```

Sample Output:

```
No. of intersection points: 4
(260.53, 409.10)
(318.94, 381.50)
(464.13, 312.91)
(521.59, 548.13)
Linear fit: 0.2937x + 297.9693
```



```
number of intersections: 4
The linear fit line is of the form: 0.2937436453278601x + 297.9692534062464
```