CSE 225L, Data Structure & Algorithm

Final Assignment

Total Marks: 25

Deadline: 04th June, 2020 (11:59 PM)

Instructions:

Read the following instructions carefully before proceeding to the questions.

1. There are only three questions but there will be partial marking. So even you do not manage to solve the whole problem, try to solve as much as you can.

2. There will be no mark for plagiarizing. If you plagiarize in any way, your submission will be rejected and you will get 0 out of 25.

3. You will be eligible for a 40% bonus of your obtained mark, if you use github.

    a. You must create a private repository for the sole purpose of submitting this assignment. If your repository is public, not only you will get no bonus but you will be penalized. Because, pushing code in a public repository can make your code exposed to others.

    b. You must give me access to your private repository before you push your codes.

    c. You must make different commits for different problems. Do not make a single commit for the whole assignment.

    d. If you do not know how to create private repository and how to give others access, contact me and I will help you with this.

4. You can submit your code in google classroom like any other assignment. Make sure that you have saved everything before submitting. If you are using github to submit assignment, then just write your repository link in a text file and submit in google classroom.

Good luck with your assignment, happy coding and

Eid Mubarak.

Maintain social distance and stay safe.

1. [10 marks] Create a class called points that has variables to store x and y co-ordinates, constructor function as well as a function to print the co-ordinate values.
   Now modify priority queue from class template to a class that works with your points class. Points that are closer to the origin (0, 0) should be considered with higher priority.

   In the driver file (main.cpp), create a few points objects, enqueue them in a priority queue then print the queue.

2. [8 Marks] Write a program to determine the height and the total number of leaf nodes of a binary search tree.

   Input to your program starts with an integer N denoting the number of elements of binary search tree followed by N integer elements.

   Your program should construct a binary search tree inserting the elements in the same order as they are given in the input and then print the number of leaf nodes and the height of your tree.

3. [7 Marks] Write a function “reverse” in your stack class that reverses the whole stack.

   In your driver file (main.cpp), create an integer stack, push some values in it, call the reverse function to reverse the stack and then print the stack.

Note: To get the data structure code, clone this repository or download it. I have fixed a few issues in the codes so codes from this repository will be best to use. If you have trouble cloning the repository or downloading it, contact me.

Repository link: https://github.com/LiquidO2/NSU-LAB