EECS 233 Assignment #2

100 points

Due October 9, 2023 before 11:59pm

General instruction: This assignment includes two parts, Written Exercises and programming exercises. Please write/type your answers neatly so they can be readable. Please submit a single PDF file for the written part and a zip file for the programming part before 11:59 P.M. on October 9, 2023.
File name format: P2_YourCaseID_YourLastName.zip (or pdf).

Written Exercises

a.Answer the following questions: (30 points)

1) It is known that each element in the list occupies 5 storage units, and the storage address of the 15th element is 350. What is the first address of the list?

2) What is the time complexity(runtime) of deleting a specified element in a singly linked list? Briefly describe the operation process.

3) In a doubly linked list, what statement should be executed to delete the successor node to the node P? Briefly describe the operation process.

4) What do stacks and queues have in common?

5) If the input sequence of a stack is ABCDE, provide three possible output sequences.

6) A binary tree with a depth of 10, what is the maximum number of nodes and the minimum number of nodes?

7) A complete binary tree has 128 nodes. What is the depth of the complete binary tree? How many leaf nodes does it have?

8) What is the total number of binary search trees that can be built with 4 different keys?

9) What is the minimum number of nodes to build an AVL tree of height 5?

10) Briefly describe binary trees, balanced binary trees, AVL trees and B-trees.

b.This question has two subparts, which follow one another. (10 points)

A binary tree has a preorder traversal sequence A E F B G C D H I K J and a inorder traversal sequence E F A G B C H K I J D.

a) Draw this binary tree.

b) Write its postorder traversal sequence.

c.This question has three subparts, which follow one another. (10 points)

1) Insert 35, 51, 43, 23, 70, 15, 37, 27, 52 to an empty AVL tree, in the given order. Show only the final tree after all insertions.

2) Add one more element such that it causes a single **left** rotation in the tree. State the added number and show the final tree after the insertion.

3) Insert 30 to the AVL tree. And show the final tree.

Programming Exercises

a.There is a doubly linked table L. Design an algorithm to find the node X and swap it with its successor node. (25 points)

1) Give code.

2) Explain the major concept of your algorithm.

3) Give an example of running your algorithm.

4) Give and justify its runtime.

b. Create a class named BinarySearchTree with the following methods: (25 points)

1) `void insert(Node root, int key)` – inserts a node in the BST.

2) `void preorderRec(Node root)` – preorder traversal of BST .

3) `int sum(Node root)` – find sum of all the keys of a BST.

4) `Node kthBiggest(Node root, int k)` – find the k'th biggest element in BST.

Submission

The submissions will be evaluated on completeness, correctness, and clarity. Please provide sufficient comments in your source code to help the TAs read it. Please generate a single zip file containing all your *.java files needed for this assignment and optionally a README.txt file with an explanation about added classes and extra changes you may have done. Name your file 'P2_YourCaseID_YourLastName.zip'. Submit your zip file electronically to Canvas.

Notice: Before your programming assignment is graded, TAs will feed other examples to your code as input, so please make sure your program works for any input before your submission. You can test your code by feeding multiple inputs and check them if they are correct.

Grading (for each programming question):

- Implementation: 7 pts

- Customized Demo with all functions included: 8 pts
- Proper encapsulation/information hiding: 2 pts
- Design and style: 4 pt
  - Style: 1 pt
  - Comments: 2 pt
  - Design: 1 pt