Homework 3

CS201: Data Structures and Algorithms, Fall 2021
Due: 4/12/2021, 11:59pm

Topic: Recurrence relations, Proof by induction, Binary search Tree

Problem 1:

Solve the following recurrence relations and prove your result by induction.

1. $T(n) = T(n/5) + 5, T(1) = 31$
2. $T(n) = 2T(n/2) + n, T(1) = 1$
3. $T(n) = T(n-1) + n^2, T(1) = 23$

Problem 2:

Draw the recursion tree for the following recurrence relations and use it to solve it. Proof is not required.

1. $T(n) = 2T(n/2) + n + 1,\ T(1) = 1$
2. $T(n) = 5T(n/5) + n,\ T(1) = 1$

Problem 3:

Create a class that implements binary search tree (BST) data structure. Your class shall include the following APIs:

1. `boolean search(int)`
2. `void delete(int)`
3. `void add(int)`
4. `void printInorder()`
5. `void printPostorder()`
6. `void printPreorder()`

Problem 4:

Write a function that counts the number of leaf nodes in a BST. Determine the time complexity of your implementation.

Problem 5:

Write a function that finds the k^{th} smallest integer in a BST. Determine the time complexity of your implementation.

Problem 6:

Write a function that checks if a binary tree is complete. Determine the time complexity of your implementation.

Bonus (15%):

Write a function that takes a BST and delete all nodes such that the result is complete BST. Determine the time complexity of your implementation.

Deliverables:

- Submit one source file for problems 3-6 and bonus.
- Add comments that explains your code. *If I can't understand ... I can't grade*
- Calculate **the time complexity** for each function in your files.
- Use only one programming Language. Either `C++,` `Java,` or `Python`
- **NOTE:** You might be selected to present your solution to the TA.

Hints and Advice:

- Start working on the homework ASAP.
- Ask questions if you feel you are lost.