Homework 3
Com S 311
Due: Feb 16, 11:59PM

Late Submissions are NOT accepted. Solutions will be posted Feb 17.

There are 4 problems and each problem is worth 60 points. When asked to design and algorithm, please write *pseudo code*. If you write Java/C code, it will not be graded.

1. Let A be an array consisting of elements in the following order

$$21, 30, 25, 36, 34, 35, 15, 17, 12$$

   (a) Draw the binary search tree that is obtained by adding/inserting the elements (without balancing) in the order they appear in the array.

   (b) If the resulting tree is not balanced, perform balancing as per AVL tree discussed in lectures. Specify the steps needed to balance (left rotation/right rotation etc) and draw the resulting balanced tree.

   (c) Draw the hash table (using chain hashing) obtained by by adding the above elements using the hash function $(4x + 1)\%5$.

   (d) For a BST T and a node N in T, let $B(N)$ = height of left sub tree of N - height of right subtree of N. Note that $B(N)$ could be positive, negative or 0. Let T be the following BST: The root node is X, it has a left subtree named R and a right subtree rooted at Y. The node Y has a right subtree Z and a left subtree rooted at node U. The node U has two sub trees V and W. Suppose that $B(U) = 0$, $B(Y) = 1$ and $B(X) = -2$. Draw the tree obtained by balancing T. Draw the intermediate trees (if any) obtained while balancing. Once you balance the tree, what are $B(X), B(Y)$ and $B(U)$?

2. Suppose that you are given Binary Search Tree T and and integer k. Assume that nodes of T hold distinct integers. Give two integers a and b ($a \neq b$) in T let

$$Distance_{a,b} = |a + b - k|$$

   Give an algorithm that gets T and k as inputs and finds a pair $a, b \in T$ such that

$$Distance_{a,b} = \min\{Distance_{u,v} \mid u \in T, v \in T, u \neq v\}.$$

   Analyze the worst-case time complexity of your algorithm and justify the correctness.

3. Design a data structure D that can store integers. The data structure should be ale to store a multi-set, i.e. an element can appear multiple times. The data structure should support the following operations in $O(\log n)$ time, where n is the number of distinct integers stored in D.

   - `add(x)`: Adds/inserts integer x into D. Even if x belongs to D, x should still be added.
   - `frequency(x)`: Number of times x appears in D.
   - `search(x)`: Returns true if x is in D.
   - `order(y)`: Number of elements in D that are larger than y. The element y may or may not be in D.

   If you are using BST and would like to balance the tree, you may simply write `balance` and assume that this operation will take $O(\log n)$ time. Analyze the worst-case run time of the operations.

4. For a set S consisting of (distinct) integers and an element $x \in S$, $rank(x)$ is the number of elements in S that are less than or equal to x. Given a Binary Search Tree T, give a recursive algorithm, that outputs all leaf nodes along with their ranks. Analyze the worst-case run time of your algorithm and prove its correctness. Your algorithm must be recursive, otherwise you will not receive any credit.

GUIDE LINES:

- Please write your recitation number, time and TA name.
- It is important to know whether your really know! For each problem, if you write the statement "I do not know how to solve this problem" (and nothing else), you will receive 20% credit for that problem. If you do write a solution, then your grade could be anywhere between 0% to 100%. To receive this 20% credit, you must explicitly state that you do not know how to solve the problem.
- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.
- When proofs are required, you should make them both clear and rigorous. Do not hand waive.
- If you hand writing is not legible, then your homework will not be graded.
- Any concerns about grading should be made within one week of returning the homework.
- **Please submit your HW via Canvas. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. Name your file must be *YourNetID-HW3.pdf*. For example, if your net id is `potterh`, then the file name should be *potterh-HW3.pdf*. HW's submitted in incorrect format (non pdf, incorrect file name etc) will incur a penalty of 20%**