Homework 3
Fundamental Algorithms, Fall 2020, Professor Yap

Due: Wed Oct 21, by 11pm in GradeScope.

INSTRUCTIONS:

- Remember that we have a "no late homework" policy. Special permission must be obtained in advance if you have a valid reason.
- Only hand in answers to the questions with positive points attached to them. Do the others at your leisure.

---

1. (6 Points) Recall that a rotation can be implemented with 6 pointer assignments. Suppose a BST maintains successor and predecessor links (denoted u.`succ` and u.`pred` in the ¶III.22, p.23). How many of pointer assignments does a rotation need in such a BST? *First specify an integer, then justify your answer.* (But be careful!)

2. (12 Points) Please do Exercise III.6.9 (p. 52) on AVL height.

3. (12 Points)
(a) Consider a linear recurrence of the form

$$T(n) = C + \sum_{i=1}^{k} a_i T(n-i)$$

where $C, a_1, \ldots, a_k$ are constants. We say the recurrence is **homogeneous** if $C = 0$, else **non-homogeneous**. If $T(n)$ is non-homogeneous, show how you can transform it into a homogeneous linear recurrence,

$$t(n) = \sum_{i=1}^{k} a_i t(n-i).$$

What is the connection between $T(n)$ and $t(n)$?
(b) Recall that the min-size AVL tree of height h satisfies the non-homogeneous recurrence

$$\mu(h) = \begin{cases} 1+h & \text{if } h = 0, 1, \\ 1 + \mu(h-1) + \mu(h-2) & \text{if } h \geq 2. \end{cases}$$

Give an exact solution for $\mu(h)$ of the form

$$\mu(h) = A\phi^h + B\widehat{\phi}^h$$

for all $h \geq 0$. You must determine the constants A, B, using the initial conditions for $\mu(0)$ and $\mu(1)$. Recall that $\phi = \frac{1+\sqrt{5}}{2} = 1.6180\ldots$ and $\widehat{\phi} = 1 - \phi = -0.6180\ldots$.

4. (20 Points) (This is a essentially Ex.III.6.10, p. 52.)
We know that a single insertion can always be fixed by a single rebalancing act. By a **rebalancing act**, we mean either a single rotation or a double rotation as performed during our Rebalance Phase. But how many rebalancing acts can be caused by a deletion?

   Let $m(k)$ be the minimum size AVL trees such that a single deletion will cause k rebalancing acts.
(a) Determine $m(1), m(2), m(3)$. You must draw the AVL tree for each $m(k)$, and mark a node whose deletion will cause k rebalancing acts.
(b) Prove that $m(k) = \mu(2k)$.
(c) Can you design the AVL tree for $m(k)$ in part(b) to be any combination of k single (S) or double (D) rotations? E.g., $k = 2$ has four combinations: SS, SD, DS, DD.
(d) In an AVL tree of size n, what is the most number of rebalancing acts you can get after a deletion. Note that $O(\log n)$ is a trivial answer, so we really what a non-asymptotic bound.

---

5. (16 Points) (a) Give a non-recursive routine called $isBST(u)$ which returns `true` if the node u represents (the root of) a BST; otherwise it returns `false`. Although $isBST(u)$ is non-recursive, it calls a *recursive* routine which we denote by $R(u)$. HINT: The subroutine $R(u)$ returns a pair of values.

   (b) Please specify clearly what is returned by $R(u)$, and give a brief proof that your routine for $R(u)$ is correct.

   Notes. Assume that a node u is either `nil`, or else it has three fields: $u.\mathtt{key}, u.\mathtt{left}, u.\mathtt{right}$. The keys k in a BST are, by definition finite, i.e., $-\infty < k < +\infty$.

6. (16 Points) The ratio bound ρ in the ratio balanced class $RB[\rho]$ is normally restricted to $(0, \frac{1}{2})$. Let us explore why we don't extend to $\rho \in [\frac{1}{2}, 1)$.
   (a) What is wrong with the class $RB[\frac{2}{3}]$?
   (b) Which other values of $\rho \in [\frac{1}{2}, 1)$ pose problem similar to part(a)?
   (c) Suppose we *really* want to allow values of ρ that lie in the range $(\frac{1}{2}, 1)$. How can we overcome the above objections?

7. (24 Points) Relaxed AVL Trees
   Let us define **AVL[2] balance condition** to mean that at each node u in the binary tree, $|balance(u)| \leq 2$.
   (a) Draw the smallest size AVL[2] trees of heights $h = 0, 1, 2, 3, 4, 5$.
   (b) Derive an upper bound on the height of a AVL[2] tree on n nodes of the form. You may imitate the arguments in our Lectures Notes for the usual AVL trees. Give the best bound you can.
   (c) Give an insertion algorithm that preserves AVL[2] trees. Try to follow the original AVL insertion as much as possible; but point out differences from the original insertion.
   (d) Give the deletion algorithm for AVL[2] trees.

8. (0 Points) This is for your own practice (don't submit). Insert into an initially empty AVL tree the following sequence of keys: $1, 2, 3, \ldots, 14, 15$.
   (a) Draw the AVL after inserting the key 15.
   (b) Prove the following: if we continue in this manner, we will have a complete binary tree at the end of inserting key $2^n - 1$ for all $n \geq 1$.