CS6033 Homework Assignment 4*

Due Tuesday, Oct 9th at 5:00 p.m.
Turn in this assignment as a PDF file on NYU classes
No late assignments accepted

1. (10 points) Insert in order the elements $4, 32, 21, 5, 54, 20, 12, 29, 19, 10$ into a $2-3-4$ tree.

   Convert the $2-3-4$ tree you constructed into a red-black binary search tree. Indicate the color of each node by using letters B (black) and R (red). You may omit representing the `NIL` nodes.

2. (10 points) Insert in order the keys $4, 32, 21, 5, 54, 20, 12, 29, 19, 10$ into a red-black BST tree. Show your tree after every insertion. Indicate the color of each node by letters B (black) and R (red). You may omit representing the `NIL` nodes.

   On your completed tree show the black height of every node.

   After that, convert the red-black BST you have constructed into a $2-3-4$ tree, using the steps we discussed in class.

3. (5 points) What is the minimum and maximum height of a $2-3-4$ tree of n nodes. Prove your answer.

4. (5 points) Show that the longest simple path from a node x in a red-black tree to a descendant leaf has length at most twice that of the shortest simple path from x to a descendant leaf.

5. (20 points) When inserting a node into a *red-black* tree there might occur a *red-red* violation. Show how each of the cases (i.e. case 1, case 2, and case 3) to solve the red-red violation can be understood by showing how it would be solved in a $2-3-4$ tree (aka $2-4$ tree)

6. Suppose your neighbor, sweet Mrs. McGregor, has invited you to her house to help her with a computer problem. She has a huge collection of JPEG images of bunny rabbits stored on her computer and a shoebox full of 1 gigabyte USB drives. She is

*Many of these questions came from outside sources. Thank you to Bill Gasarch & Brian Perry for inspiration.

asking that you help her copy her images onto the drives in a way that minimizes the number of drives used. It is easy to determine the size of each image, but finding the optimal way of storing images on the fewest number of drives is an instance of the bin packing problem, which is a difficult problem to solve in general. Nevertheless, Mrs. McGregor has suggested that you use the first fit heuristic to solve this problem, which she recalls from her days as a young computer scientist. In applying this heuristic here, you would consider the images one at a time and, for each image, I, you would store it on the first USB drive where it would fit, considering the drives in order by their remaining storage capacity. Unfortunately, Mrs. McGregor's way of doing this results in an algorithm with a running time of $O(mn)$, where m is the number of images and $n < m$ is the number of USB drives. Describe how to implement the first fit algorithm here in $O(m \log n)$ time instead.

Question from: Goodrich, Michael T.; Tamassia, Roberto. Algorithm Design and Applications

7. Imagine that you work for a database company, which has a popular system for maintaining sorted sets. After a negative review in an influential technology website, the company has decided it needs to convert all of its indexing software from using sorted arrays to an indexing strategy based on using binary search trees, so as to be able to support insertions and deletions more efficiently. Your job is to write a program that can take a sorted array, A, of n elements, and construct a binary search tree, T, storing these same elements, so that doing a binary search for any element in T will run in $O(\log n)$ time. Describe an $O(n)$-time algorithm for doing this conversion.

   Question from: Goodrich, Michael T.; Tamassia, Roberto. Algorithm Design and Applications

8. Add an attribute to each node in a binary search tree, T, that holds the *size* of the node. The size of a node t, `size(t)`, is the number of nodes in the subtree rooted at t. (i.e. the size of a node t is $1 + \texttt{size}(t.right) + \texttt{size}(t.right))$

   For this class the definition of a balanced tree is a tree whose height is $O(\log n))$.

   - (5 points) Show that a binary search tree is balanced if $\texttt{max}\{\texttt{size(t.left), size(t.right)}\} \leq \frac{2}{3}\,\texttt{size}(t)$ for every node t in the tree.
   - (5 points) In class, when discussing red-black tree in class, we discussed the operations `left-rotate` and `right-rotate`. Write the code for these operations so that after a rotation all the nodes maintain the `size` attributes. (You may use the code given in the lecture 4 slides in your answer.)
   - (10 points) Write the pseudo code to perfectly balance a tree, T, using rotations. The nodes have the `size` attribute. We will call a tree *perfectly* balanced if for

every node $t \in T$ then `size(t.right), size(t.left)` $\leq \lfloor$ `size`$(t)/2\rfloor$. (Hint: The median value must be the root of each subtree.)

9. The party at the Chancellor's house is about to start.

   The party is a time for fun, but it could also spell calamity if the Chancellor has too much space juice and forgets a name, or mistakes a person's planet for the tabloid featured planet, Kardashian. That person may decide "This means war!"

   To prevent the outbreak of war, design a system to help the Chancellor navigate the treacherous waters of small talk.

   Your system will perform the following:

   - Insert a new name (and the person's home planet, alliances, favorite color and kompromat) to the guest list in *worst* case time $O(\log n)$ where n is the size of the guest list
   - Return the home planet, alliances, favorite color, and kompromat of a person when provided the person's name in $O(\log n)$ *worst* case time
   - Delete[1] a name from the guest list in *worst* case time $O(\log n)$
   - When given a letter of the alphabet, return all the names in the guest list starting[2] with that letter in *worst* case time $O(g + \log n)$, where g is the number of people whose name fits the search criteria.

10. The federation is running out of money[3]. In an effort to increase revenue, the federation has decided to put toll booths on the edges of black holes; the toll will be based on the spaceship weight.(Brilliant right!)

    The toll booths (aka black hole locations) have been numbered $h_1, h_2, \ldots, h_n$ according to how close they are to federation headquarters[4] where ties are broken arbitrarily.

    Design a data structure and implement methods that can perform the following in $O(\log n)$ *worst* case time.

    (a) insert(d, r) Inserts a new toll booth which has a toll rate of r per ton. The new toll booth is at distance d from the federation headquarters.

    (b) delete(k) Deletes the k^{th} toll booth

    (c) toll(i, j) Returns the average toll for booths i through j inclusive.

---

[1]Life is so unfair, alliances shift and someone gets uninvited.

[2]Even if the Chancellor cannot remember a person's name, the Chancellor always seems to remember the first letter of a person's name.

[3]You can't be blamed, anyone would have thought the blinking light at the Chancellor's party was the juke box. Its too bad the nuclear warhead you launched took out a nearby planet and nearly bankrupted the entire federation who had to pay reparations.

[4]i.e. It is important to know which are closest when needing funds to be sent quickly.

Justify the running times of your methods.

11. (3 bonus points) Think of a *good*[5] exam question for the material covered in Lecture 4.

[5] Only well thought out questions will receive the bonus points.