CS6033 Homework Assignment*4

Due Feb. 21st at 5:30 p.m.
Turn in this assignment as a PDF file on NYU classes
No late assignments accepted

1. (10 points) Suppose you are hired as a consultant to a professor, Dr. Bob Loblaw, from the Sociology department. He is asking that you build him a software system that can maintain a set, P, of people from a country, Phishnonia, that he is studying. People in Phishnonia are free to come and go as they please, so Dr. Bob Loblaw is asking that your system support fast insertions and deletions. In addition, he is also interested in doing queries that respectively return the mean and median ages of the people in P. His thesis is that if the median age is much smaller than the mean, then the Phishnonia is ripe for revolution, for it implies that there are a disproportionate number of young people. Describe a scheme for maintaining P in a binary tree that can support median and mean age queries, subject to insertions and removals, with each of these operations running in at most $O(logn)$ time, where n is the number of people in P.

2. (10 points) Insert the letters `a,l,g,o,r,i,t,h,m,s` into an initially empty red-black tree.[1] Show the tree after:

   - every insertion before fixing up the tree. If there is a violation, label the violation as a case 1, 2, or 3 violation
   - every resolution of a violation. If there is a new violation, label the violation as a case 1, 2, or 3 violation

3. (10 points) Insert the letters `a,l,g,o,r,i,t,h,m,s` into an initially empty 2-3-4 Tree. Show the tree after every insertion.

4. (10 points) What is the largest possible number of internal nodes in a red-black tree of height h? What is the smallest possible number of internal nodes of a red-black tree of height h?

*Many of these questions came from outside sources.
[1]Given two distinct letters l_1 and l_2, will say $l_1 < l_2$ if l_1 comes before l_2 in the alphabet.

5. (20 points) In class we augmented the red-black tree so each node included it's size. Suppose we instead stored in each node its rank in the subtree of which it is the root. (The size of each node is not stored.)

   - Show how to maintain this information during insertion
   - Write the methods `OS-SELECT(x, i)`
   - Write the methods `OS-RANK(T, x)`
   - State the *worst case* run time of inserting an item, `OS-SELECT`, and `OS-RANK` using Theta notation.

   Which is more efficient: augmenting the data structure by adding at each node the size of the subtree of which it is the root, or augmenting the data structure by adding at each node the rank of that node of the subtree of which it is the root?

6. Implement a data structure that supports the operations: *INSERT* and *TOTAL* in $O(\log(n))$ time. This data structure will keep track of the number of movie tickets sold on a specific day. Assume that the number of days recorded could be very large.

   (a) (20 points) INSERT(T, date, number of people). This operation inserts the number of tickets sold on a specific date. (The dates might not be inserted in order.)

   (b) (20 points) TOTAL(T, date1, date2). This operation returns the total number of tickets sold between the dates date1 and date2 (including date 1 and date 2).

   (c) (20 bonus points) DELETE(T, date). This operation[2] deletes the information recorded on the specified date. It must run in $O(\log(n))$ time.

   Describe how your data structure handles insertions, and finding the number of tickets sold between two dates by writing the pseudo code for INSERT(T, date, number of people), and TOTAL(x, date1, date2).

   In your answer, you may modify any existing data structure/algorithm we have discussed in class as long as you *describe any modifications* your algorithm needs. If you modify an existing data structure/algorithm, you do not need to rewrite the code. Instead, write the *new lines* of code you would add to the existing code, and *state where the new lines belong.*

7. (3 bonus points) Think of a *good*[3] exam question for the material covered in Lecture 4

[2] We didn't cover this topic in class - please look in the book to learn how to delete an item from a red-black tree.

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