Ann, Bob, Charlie (*replace with your names and correct the date*)
COSC 336
3/19/2500

Assignment 6

Instructions.

1. Due date: As indicated on Blackboard..
2. This is a team assignment. Work in teams as in the previous assignments. Submit one assignment per team, with the names of all students making the team.
3. Your programs must be written in Java.
4. Write your programs neatly - imagine yourself grading your program and see if it is easy to read and understand. At the very beginning present your algorithm in plain English or in pseudo-code (or both). Comment your programs reasonably: there is no need to comment lines like "i++" but do include brief comments describing the main purpose of a specific block of lines.
5. You will submit on Blackboard 2 files. The first file should be a .pdf file with the solution to the exercises and with descriptions in English or in pseudocode of the algorithms for the programming task you are required to do and the results that you are required to report. Make sure you label the results as indicated below. Insert images/screenshots with the output you obtain for each testing data. Also include the code of your program so that the grader can make comments. The second file will contain the Java sources of your two programs.

   For editing the above document with Latex, see the template posted on the course website.

   assignment-template.tex and

   assignment-template.pdf

   To append in the latex file a pdf file, place it in the same folder and then include them in the latex file with

   ```
   \includepdf[pages=-,pagecommand={},width=\textwidth]{file.pdf}
   ```

   To append in the latex file a .jpg file (for a photo), use

   ```
   \includegraphics[width=\linewidth]{file.jpg}
   ```

Exercise 1. Recall the Partition subroutine employed by QuickSort. You are told that the following array has been partitioned around some pivot element:

3	1	2	4	5	8	7	6	9

Which of the elements could have been the pivot element? (List all that apply; there could be more than one possibility.)

Exercise 2. Let α be some constant, independent of the input array length n, strictly between 0 and 1/2. What is the probability that, with a randomly chosen pivot element, the Partition function produces a split in which the size of both the resulting subproblems is at least $\alpha \cdot n$. Choose the answer from the following list and justify your answer.

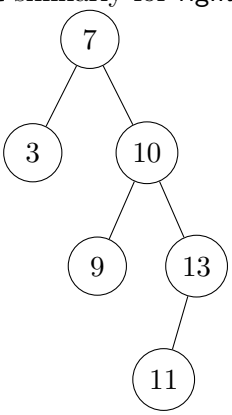
- α
- $1 - \alpha$
- $1 - 2\alpha$
- $2 - 2\alpha$

Programming task Your task is to augment the class BinarySearchTree from https://www.geeksforgeeks.org/binary-search-tree-set-1-search-and-insertion/

Modify the insert function member so that duplicates are also inserted (which is not done in the version at the above website). If the value x to be inserted is equal to the value in the root, insert x in the left subtree.

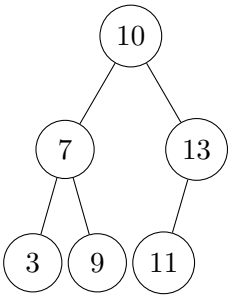
First, you'll add to the class Node a data member called int size which keeps the number of nodes in the tree rooted at that node (including in the count the node itself). The constructors and the insertion function need to take into account the sizes of the nodes.

Second you'll add to the class BinarySearchTree, two function members called leftRotate (Node t), which rotates the root t to the left, so that the right child of t (if there is one; otherwise the rotation does not do anything) becomes the parent of t, and symmetrically rightRotate (Node t). See Figure 13.2, page 313 in the textbook, or Notes 6 on Blackboard. Note that when you do leftRotate (Node t), you need to change the size of t and of t.right, and similarly for rightRotate.



For instance for the tree in the figure, node 7 has size 6, node 3 has size 1, node 9 has size 1, node 10 has size 4, node 13 has size 2, and node 11 has size 1.

If we do leftRotate for the root we get the tree



To test the augmented class, you will insert some values and then print in the *preorder* order the pairs (value, size) of all the nodes. Next, you leftRotate the root and print again the tree after rotation.

Test data 1: insert 7, 10, 3, 9, 13, 11. Your program will print: (7,6), (3,1), (10,4), (9,1), (13, 2), (11,1). Next, do a leftRotate, and print the tree after rotation and you get (10,6), (7,3), (3,1), (9,1) (13,2), (11, 1).

Test data 2: Insert one by one the numbers in the file input-6.1. The first line contains how many numbers are there (there are 1000 numbers), and the next line contains the list

of numbers.

Report in the pdf file the first 25 pairs.

Test data 3: Insert one by one the numbers in the file input-6.2. The first line contains how many numbers are there (there are 10000 numbers), and the next line contains the list of numbers.

Report in the pdf file the first 25 pairs.