Name_____

CSCI 151
Practice Midterm 2
December 2021

This practice midterm has many questions for you to use to study for the exam. Note, this exam is way longer than the exam Professor Geitz is going to give, we have done that intentionally so that you have many questions to practice with.

Good Luck! You got this!

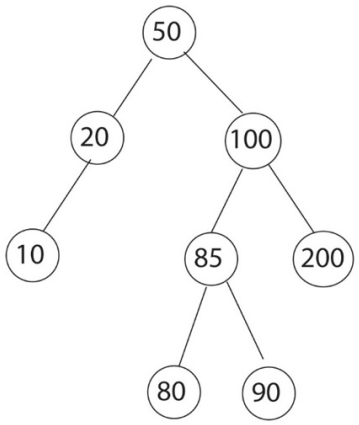
1. S is a structure of type MysteriousStructure. All that you know about S is that it has an iterator method S.iterator( ) that returns a value of type Iterator<Integer>, and this iterator goes through all of the values stored in S. Write function

   Integer largest(MysteriousStructure S)

   that returns the largest value in S, or null if S is empty.

```
Integer largest(MysteriousStructure S) {
    Iterator<Integer> it = S.iterator();
    Integer big;
    if (!it.hasNext() )
          return null;
    else
          big = it.next();
    while (it.hasNext() ) {
          Integer x = it.next();
          if (x > big)
              big = x;
    }
    return big;
}
```

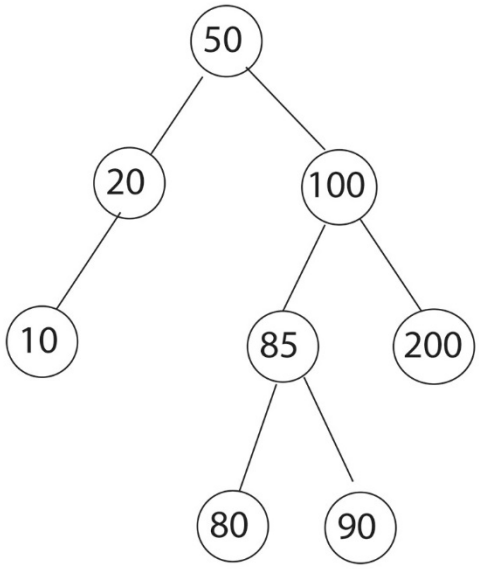
2. Here is a picture of a binary tree with an integer stored in each node.



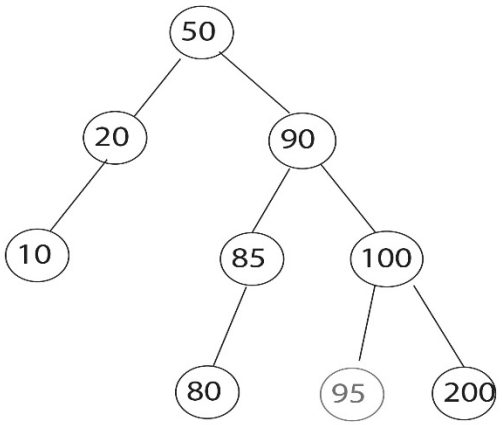
   List the node values in pre-order, in-order and post-order traversals of this tree. Label your traversals so I can tell which is which.

pre-order: 50 20 10 100 85 80 90 200
in-order: 10 20 50 80 85 90 100 200
post-order: 10 20 80 90 85 200 100 50

3. Here is an AVL tree



and here is a "level-by-level" listing of its node values: 50 20 100 10 85 200 80 90 (just taking the node at level 0, then the nodes at level 1, etc., each level listed from left to right). **Find the AVL tree that results from adding value 95 to this tree** and give its level-by-level listing. [Note: I am asking for this level-by-level listing, so you don't need to draw the tree; if you find it easier to draw the tree, do that rather than give the listing.



First, here is the tree with the added node. Z is the first node up from the addition that fails the AVL property; Y is Z's tallest child; X is Y's tallest child. a, b, and c are X, Y and Z in increasing order. h1 through h4 are the four children of X, Y, and Z (who are not X, Y, and Z) from left to right.

4. Describe a non-recursive algorithm for an inorder tree traversal (use a stack)

1) Create an empty stack S.
2) Initialize current node as root
3) Push the current node to S and set current = current->left until current is NULL
4) If current is NULL and stack is not empty then
   a) Pop the top item from stack.
   b) Print the popped item, set current = popped_item->right
   c) Go to step 3.
5) If current is NULL and stack is empty then we are done.

5. Write code to find the minimum and maximum element in a binary search tree. If its height is *H*, what is your algorithm's runtime?

Minimum: traverse down the left child until reach leaf. Runtime O(h)
Maximum, traverse down right child until reach leaf. Runtime O(h)

6. Draw out the steps of sorting the following array of integers using merge sort [15, 3, 12, 5, 7, 1, 8, 11]

[15,3,12,5] [7,1,8,11]
[15,3] [12,5] [7,1] [8,11]
[15] [3] [12] [5] [7] [1] [8] [11]
[3,15] [5,12] [1,7] [8,11]
[3,5,12,15] [1,7,8,11]
[1,3,5,7,8,11,12,15]

7. Give an example of a worst case input to mergesort.

Trick question, there is no worst case.