Name: ______________________________ Andrew Id: ____________________

15-121 Fall 2022 Quiz 9

Up to 20 minutes. Show your work. No calculators, no notes, no books, no computers, no other people.

1. (8 points) **Hash Tables**

   Complete the following code for a hash table. When determining how many buckets you need, you should have 1.3x the expected number of items. Also, you need to ensure that duplicate items are never added to the table. (If there is an attempt to add a duplicate item, then do not add it.)

```java
public class MyHashtable<DataType> {
    private ArrayList<DataType>[] buckets;

    public MyHashtable(int numItems) {











    }

    public void add(DataType item) {











    }

    public boolean contains(DataType item) {











    }
}
```

2. (12 points) **Non-Recursive BST Iteration**

In class we discussed that one issue with binary search trees is that they are difficult to iterate over without using recursion. One potential solution to this problem is modify a basic BST so that every node is both part of the tree and also part of a linked list. Then, if you need to iterate over all the items in the tree, you can simply follow the linked list.

Consider the following code for a hybrid data structure, the linked list binary search tree:

```java
public class LinkedListBST<TheType extends Comparable<TheType>> {
    private Node head;
    private Node root;

    private class Node {
        private TheType data;
        // Left and right pointer for the tree this node is in
        private Node left;
        private Node right;
        // Next pointer for the linked list this node is in
        private Node next;

        public Node(TheType item) {
            this.data = item;
        }
    }

    public void add(TheType item) {
        this.root = add(this.root, item);
    }

    private Node add(Node node, TheType item) {
        if (node == null) {
            // Create node, add it to the LL, and return it so it gets added to the BST
            Node ret = new Node(item);
            ret.next = head;
            head = ret;
            return ret;
        }
        int res = item.compareTo(node.data);
        if (res < 0) {
            node.left = add(node.left, item);
        } else if (res >= 0) {
            node.right = add(node.right, item);
        }
        return node;
    }
}
```

Right now, trying to use a for-each loop with `LinkedListBST` does not work. Java reports the following error:
`Can only iterate over an array or an instance of java.lang.Iterable`

Modify the `LinkedListBST` so that for-each loops work properly on it.

Note: If you need to modify the existing code for the class, make those changes on this page. If you need to add new things to the class, then write them on the next page.

Space for answer to Question 2