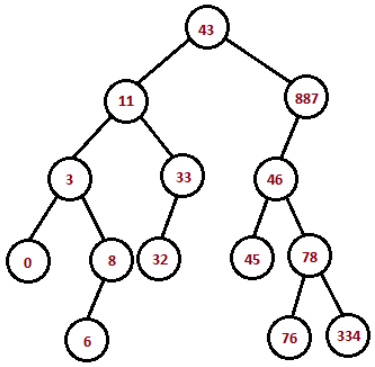
[**Closed book and notes.**] Show all of your work clearly in the space provided. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (10 points) Suppose you are asked to implement a Stack data structure using either an ArrayList or a LinkedList. Is there a significant reason for using one in favor of the other? Justify your answer.

2. (5 points) Explain the rules governing the size comparison between a parent and its child nodes in a binary search tree.

3. (5 points) Why is maintaining such an order (as described in your answer to **2.**) important for the performance of the algorithms associated with a binary search tree?

4. Consider the following binary search tree:



(a) (10 points) Suppose that the inner node class has the same attributes as the BST class developed in lecture and that the root refers to the root node in the tree. What will be displayed to the console if the following method (which is added to the BST class) were called with: print(root); for the tree in the diagram above?

```
public void print(Node node) {
  if(node!=null) {
    System.out.println(node.value);
    print(node.lKid);
    print(node.rKid);
  }
}
```

(b) (5 points) Use big-oh notation to describe the overall worst case time complexity for the print() method where n is the size of the binary search tree.

(c) (15 points) Implement a recursive height(Node node) method that returns the height of the subtree where node is the root of the subtree. Assume that if node==`null` then the subtree is empty and the height is 0.

5. Suppose that a CircularQueue class is partially defined as follows:

```
public class CircularQueue<E> {
  private Object[] buffer = new Object[512];
  private int front = 0;
  private int back = 0;
  private boolean isFull = false;
```

(a) (15 points) Implement the push() method for the class.

(b) (15 points) Implement the pop() method for the class.

6. Recall the doRecursive4WaySearch method from the word search lab assignment. Assume that the recursive component was implemented as:

```
doRecursive4WaySearch(row+1, col);
doRecursive4WaySearch(row,   col+1);
doRecursive4WaySearch(row-1, col);
doRecursive4WaySearch(row,   col-1);
```

Assume the following grid of letters as input:

A	B	C
E	F	G
H	I	J

For partial credit, be sure to explain your answer.

(a) (5 points) Suppose that the method starts at **A**, i.e., doRecursive4WaySearch(0, 0); What will currentWord contain the first time it is four characters long?

(b) (5 points) Suppose that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain the first time it has maximum length?

(c) (5 points) Now suppose that the recursive component is changed to:

```
doRecursive4WaySearch(row,   col-1);
doRecursive4WaySearch(row-1, col);
doRecursive4WaySearch(row,   col+1);
doRecursive4WaySearch(row+1, col);
```

Use same grid:

A	B	C
E	F	G
H	I	J

and that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain the **SECOND** time it is four characters long?

(d) (5 points) Now suppose that the recursive component is changed to:

```
doRecursive4WaySearch(row-1, col-1);
doRecursive4WaySearch(row-1, col+1);
doRecursive4WaySearch(row+1, col-1);
doRecursive4WaySearch(row+1, col+1);
```

and that the method starts at **F**, i.e., doRecursive4WaySearch(1, 1); What will currentWord contain the first time it has maximum length?