[**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) Enumerate and explain the methods that are part of a pure stack interface.

2. (10 points) Describe the design flaw found in the java.util.Queue interface.

3. (10 points) Use big-O notation to describe the overall worst case time complexity for a recursive method that calculates the height of a complete binary tree (e.g., the height(Node) method we implemented in lecture). Be sure to explain your reasoning.

4. (10 points) Use big-O notation to describe the overall worst case time complexity for a non-recursive method that returns the largest element in a complete binary search tree. Be sure to explain your reasoning.

5. (10 points) Suppose that the CircularQueue class has been implemented and the the integer passed to the constructor sets the capacity for the circular queue. Give the asymptotic time complexity for each of the following methods: offer(), poll(), and peek(). Justify your answers.

6. (10 points) Explain how a circular queue differs from a standard queue.

7. (a) (15 points) Implement a recursive version of the max method for a binary search tree that returns the largest element stored in the tree. Assume that an empty tree has one attribute, root, which is set to `null` when the tree is instantiated and that the recursive version is called as follows:

```
public E max() {
  return max(root);
}
```

The method should return `null` if the tree is empty.

(b) (5 points) What is the asymptotic time complexity for this method? Justify your answer.

8. (a) (15 points) Implement a recursive version of the max method for a binary tree (NOT a binary *search* tree) that returns the largest element stored in the tree. Assume that an empty tree has one attribute, root, which is set to `null` when the tree is instantiated and that the recursive version is called as follows:

```
public E max() {
  return max(root);
}
```

You may assume that elements in the tree implement the Comparable interface. The method should return `null` if the tree is empty.

(b) (5 points) What is the asymptotic time complexity for this method? Justify your answer.

Additional work area for any problem. Clearly identify which problem is associated with the work on this page.