

[**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) What is the worst-case asymptotic time complexity of for the `offer()` and `remove()` methods for a `Queue` that is implemented with a `LinkedList`? Justify your answer.

2. (10 points) Suppose you are asked to implement a `CircularQueue` with a capacity of 512 elements using either an array or an `ArrayList`. Is there a significant reason for using one in favor of the other? Justify your answer.

3. (10 points) For arbitrarily long data structures, it makes more sense to implement the `Queue` interface using a `LinkedList` instead of an `ArrayList` as the fundamental data structure on which the implementation is built. Explain why this is the case.

4. (a) (20 points) Implement a method, `parenChecker()`, that accepts a `String` and returns `true` if the parenthesization is legal. The method must make use of a `Stack` with the traditional methods found in a stack. For example:

- $((a + b) + c)$ – Good
- $([a + b] + c)$ – Good
- $(a + \{b + c\})$ – Good
- $((a + b) + c$ – Bad
- $([a + b] + c]$ – Bad
- $(a + \{b + c\}$ – Bad

Part **(a)** cont. . .

(b) (5 points) Use big-oh notation to describe the overall worst case time complexity for your algorithm where n is the length of the string passed to the method. Justify your answer.

5. (20 points) Implement a recursive version of the `contains` method for a binary search tree. Assume that a tree does not contain `null` elements, and, when empty, has a `root==null`. The method you are to implement is called as follows:

```
public boolean contains(E target) {  
    return contains(root, target);  
}
```

6. (a) (20 points) Implement a non-recursive version of the `contains` method for a binary search tree. Assume that a tree does not contain `null` elements, and, when empty, has a `root==null`. The method you are to implement is called as follows:

```
public boolean contains(E target) {  
    return contains(root, target);  
}
```

(b) (5 points) Assuming the binary search tree is **complete**, what is the worst-case asymptotic time complexity of for the `contains` method implemented in part **(a)**? Justify your answer.



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