

[**Closed book and notes.**] Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. Consider the following method:

```
public static void makeOdd(List<Integer> numbers) {
    if(numbers==null) {
        throw new IllegalArgumentException("List_cannot_be_null");
    }
    for(int i=0; i<numbers.size(); ++i) {
        if(numbers.get(i)%2==0) {
            numbers.set(i, numbers.get(i)+1);
        }
    }
}
```

(a) (10 points) What is the worst case asymptotic time complexity for makeOdd when numbers is a java.util.ArrayList. Justify your answer.

(b) (10 points) What is the worst case asymptotic time complexity for makeOdd when numbers is an java.util.LinkedList. Justify your answer.

1. continued

(c) (5 points) Suppose that the method signature is changed to:

```
public static void makeOdd( Collection<Integer> numbers) {
```

what difference would this make?

2. (10 points) Give one one advantage and one disadvantage of a singly linked list relative to a doubly linked list.

3. Recall that our implementation of the `ArrayList<E>` from lecture contains one attribute: `data` which is a reference to an array that stores all of the elements. Note that the length of the array is always the same as the size of the `ArrayList`.

(a) (20 points) Implement the `add(int index, E element)` method so that it is consistent with the class developed in lecture.

(b) (15 points) Implement the `indexOf(Object target)` method for the `ArrayList<E>` developed in lecture.

4. (10 points) Explain why the `getDesiredDots()` method required considerably more time to remove a significant number of dots than the `getDesiredDotsItr()` method when using a `LinkedList` in Lab 3.

5. (20 points) Suppose that the `tail` attribute is removed from the `LinkedList` class developed in lecture. As a result, there is only one attribute in the `LinkedList` class: `head` which points to the first element in the list.

Implement the `add(E element)` method for the modified `LinkedList`.



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