

Computer Science 62

Bruce/Mawhorter – Fall '16

Midterm Examination

October 5, 2016

Question	Points	Score
1	15	_____
2	10	_____
3	10	_____
4	8	_____
5	9	_____
TOTAL	52	_____

Your name (Please print)

1. Suppose you are given a singly-linked list class that holds strings and that maintains pointers to both the head and the tail of the list. Its fields and constructors are as follows:

```
public class SinglyLinkedList {
    protected ListNode head;
    protected ListNode tail;

    public SinglyLinkedList() {
        this.head = null;
        this.tail = null;
    }

    ...
}
```

The `ListNode` class looks like this:

```
public class ListNode {
    private String value;
    private ListNode next;

    public ListNode(String value, ListNode next) {
        this.value = value;
        this.next = next;
    }

    public String getValue() {
        return this.value;
    }

    public ListNode getNext() {
        return this.next;
    }

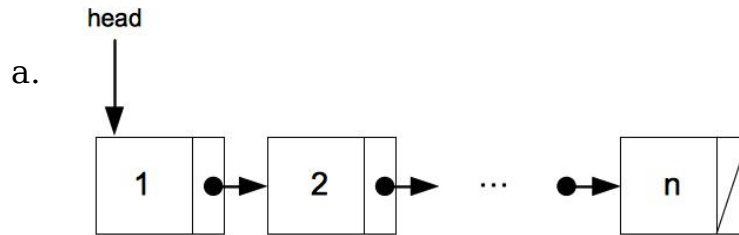
    public String setValue(String newValue) {
        this.value = newValue;
    }

    public ListNode setNext(ListNode newNext) {
        this.next = newNext;
    }
}
```



```
public void keep(int howMany) {
```

2. You have a singly linked list with only a `head` pointer (see the figure below). The `insert()` method for the list inserts new values into the list so that the elements remain in sorted order using the obvious algorithm. In other words, after each



insertion, the list is in sorted order. Assume you are given a sequence of n values to insert one at a time into

the list. What do you expect the total worst-case running time to be, using big-O notation, for inserting all of the values into the list? Give a brief (one to two sentence) **justification** for your answer.

- b. Suppose that the sequence of n values to be inserted just happen to be **in reverse sorted order**. E.g., you might be given the elements 47, 23, 19, 13, 7, 6, and finally 2. What do you expect the running time to be, using big-O notation, for inserting all of n values into the list? Give a brief (one to two sentence) **justification** for your answer.

3. Suppose we have a list of ints held as an ArrayList. As you may recall, the elements of the list are held in an array `elts`, with instance variable `eltsFilled` keeping track of the number of elements in the list. The standard implementation of an iterator for an ArrayList presents the elements of the array (one at a time) from slot 0 to slot `eltsFilled - 1`.

However, for a particular application we want to design a special iterator (call it `inOrderIterator`) that presents the elements of the list in numeric order. Please describe in words (no Java code necessary) how you would initialize the iterator (i.e., what the constructor would do), and how you would write the `hasNext()` and `next()` methods.

You may assume that the iterator is an inner class of the ArrayList class (thus giving it access to the instance variables of ArrayList). The initialization of the iterator should take no more than $O(n \log n)$ time in the worst case (where n is the size of the list) and the methods `next()` and `hasNext()` should each be $O(1)$. The contents of the array `elts` should not be modified by the construction or use of the iterator. You may use any extra storage needed to create the iterator.

4. An advantage of using stacks and queues is that their limited number of operations allows more efficient representation than more general data structures. Please answer the following questions about the complexity of operations on queues, expressing all answers in big-O notation.

A queue may be represented by a “circular” implementation on an array (or ArrayList) or by a singly linked list with a reference to both the front and rear. Please give the complexity of the following queue operations on a queue of size n for each representation, including one or two sentences justifying your answer.

i) Enqueuing an element at the rear of the queue with a

circular array:

linked list:

ii) Dequeuing an element from the front of the queue with a

circular array:

linked list:

