CSIS10B Practice Exam 2

Open Book, Closed Computer Chapters 2 – 14 (More or less)

The following topics will be emphasized:

- bag, list array implementations
- bag, list linked implementations
- stack, queue implementations
- algorithm analysis, Big-O
- basic sorting
- recursion

1) (10 pts) Big-O: If Task T in the algorithm below takes constant time to complete, what is the Big-O of these two algorithms?

a) __O(n^2)________

```
for (x = 1; x<=n; x++) {
  for (y = 1; y<=n; y++) {
    for (z = 1; z<=10; z++) {
          Task T
    }  // end for
  }  // end for
}  // end for
```

b) ___O(n log(n))_______

```
for (x = 1; x<=Math.log(n); x++) {
  for (y = 1; y<=n; y++) {
    for (z = 1; z<=10; z++) {
          Task T
    }  // end for
  }  // end for
}  // end for
```

2) (15 pts) Recursion

a) Given foo defined below, what value of n results in the base case?__when n = 4_____

And, what is printed by the following statement? __4________

System.out.println(foo(2));

```
public static int foo(int n)
{
    if (n < 4)
        return 1 + foo(n*3);
    else if (n>4)
        return 1 + foo(n/2);
    else
        return 0;
}
```

```
Foo(2)                                          done
Rtrn: 1+foo(6)                                   (4)
         Rtrn: 1+foo(3)                          (3)
                   Rtrn: 1+foo(9)                (2)
                            Rtrn: 1 + foo(4)     (1)
            Base case:                  Rtrn:    (0)
```

b) What will the following change method return if we execute the call change("xerox"); __yeroy__

```
public String change(String str) {
    if ( str.length()<1)
        return str;
    else if (str.charAt(0)=='x')
        return "y" + change(str.substring(1));    // substring(1) gives the string minus first letter
    else return str.charAt(0)+change(str.substring(1));
}
```

What is the order of growth of the change method? __O(n)______________

c) Consider the method **displayRowOfCharacters** that displays any given character the specified number of times on one line. For example, the call **displayRowOfCharacters('*', 5);** produces the line

```
*****
```

Implement this method in Java by using recursion, where only one letter is printed during each recursive call.

```
public void displayRowOfCharacters(char c, int n)
{
     if (n > 0)
    {
          System.out.print( c );
          displayRowOfCharacters( c, n-1 );
    }
}
```

3) (15 pts) Sorting

a. Given the following array:

4 15 8 3 28 21

If one pass of a sorting algorithm represents one iteration of the outer loop of the algorithm,

which of the following represents the array after the SECOND pass of a selection sort? c__

which of the following represents the array after the SECOND pass of a bubble sort? _d___

which of the following represents the array after the SECOND pass of an insertion sort? __a__

```
a) 4  8 15  3 28 21
b) 4 15 21  3  8 28
c) 3  4  8 15 28 21
d) 4  3  8 15 21 28
```

b. For the following Song class, write a compareTo method allowing objects to be compared by title:

```
public class Song
{
    String title;
    String artist;

    …

    public int compareTo( Song other)
    {
         return this.title.compareTo(other.title);
     }
}
```

c. what else must be added to the Song class so that the Java compiler will allow an array of Song objects to be sorted?

implements Comparable<Song>   at the top of class header

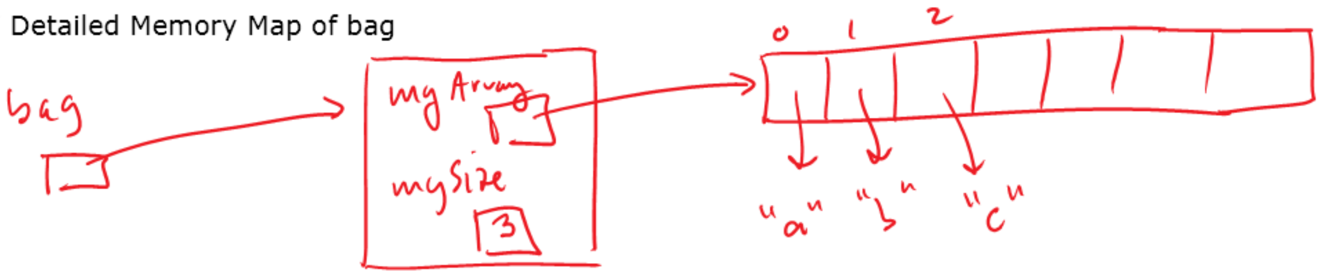
4) (20 pts) ArrayBag Class

We wish to define a new method for the ArrayBag class called addAll. This method takes an array of objects and adds them to the bag. To illustrate how it should modify the ArrayBag object, draw a before and after picture showing what happens when the addAll method is invoked on bag.

BEFORE

```
ArrayBag bag = new ArrayBag();
bag.add("a"); bag.add("b"); bag.add("c");
```

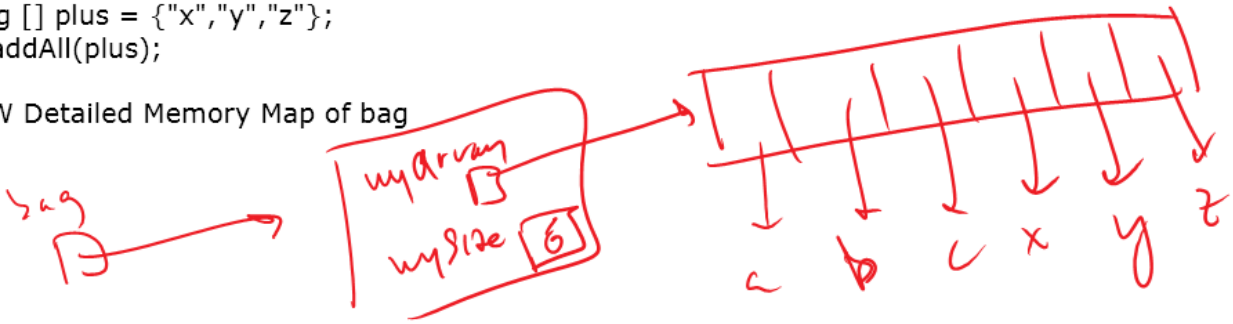
DRAW Detailed Memory Map of bag



AFTER:

```
String [] plus = {"x","y","z"};
bag.addAll(plus);
```

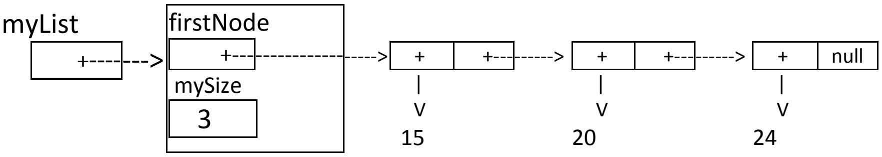
DRAW Detailed Memory Map of bag



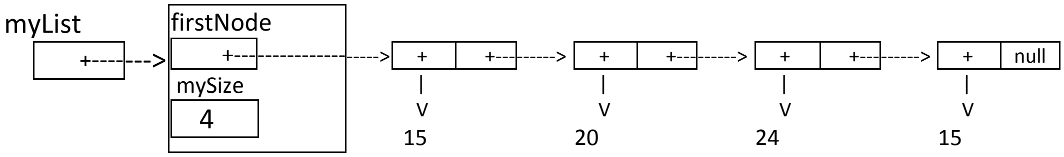
5) (30 pts) LinkedList Class Method

The method firstLast takes the first item in the list and makes a duplicate node with the same data at the end.

Here is a BEFORE picture



And here is an AFTER picture after the statement   myList.firstLast() is executed



Complete the method firstLast begun below. Use the private Node class declared at bottom.

```
public class LinkedList {
      private Node firstNode;
      private int mySize;

      public void firstLast()
      {
            //a) create newNode so it refers to a node with first item data inside

            Node newNode = new Node(firstNode.element, null);

            //b) make currentNode point to last node in the chain
            Node currentNode = firstNode;

            while( currentNode.next!= null)
                  currentNode = currentNode.next;


            //c) add newNode to the end of the list
            currentNode.next = newNode;

            //d) do any additional steps to make the list instance variables
                  consistent

            mySize++;


      }

      private class Node {
            public Object element; // The element within this Node
            public Object next; // The reference to the next node
            /**
            * Creates a new Node.
            * @param e the element within
            * @param n the next node
            */
            public Node(Object e, Node n) {
                  element = e;
                  next = n;
            }
      }
}
```