

Data Structures 1 – ST0245

First Midterm Group 034 (Tuesdays)

Name _____
Departamento de Informática y Sistemas
Universidad EAFIT

September 3, 2019

1. Give the exact output produced by the `main()` method of `Midterm` class shown below (8pts).

```
public class Counter {
    private int amt;

    public Counter(int number) {
        amt = number;
    }

    public String toString() {
        return amt + "";
    }

    public void incr() {
        amt++;
    }
}

public class Midterm {
    public static void doIt(Counter c1, Counter c2) {
        c1.incr();
        c1 = new Counter(7);
        c2.incr();
        c1 = c2;
        c1.incr();
    }

    public static void main(String[] args) {
        Counter c1 = new Counter(3);
        Counter c2 = new Counter(5);
        doIt(c1, c2);
        System.out.println("Line 1: " + c1 + " " + c2);
        doIt(c2, c1);
        System.out.println("Line 2: " + c1 + " " + c2);

        c1 = new Counter(7);
        c2 = new Counter(8);
        doIt(c1, c1);
        System.out.println("Line 3: " + c1 + " " + c2);
        doIt(c2, c2);
        System.out.println("Line 4: " + c1 + " " + c2);
    }
}
```

Output: Line 1:

Line 2:

Line 3:

Line 4:

- 2. What is the output of the main() method in the following program that calls the recursive method someMethod() (10pts)**

```
public class Recursive {
    public static int someMethod(char[] arr, int pos) {
        if (pos == 0) {
            if (arr[0] == 'X') return -1;
            if (arr[0] == 'Y') return 2;
            return 0;
        } else {
            int temp = someMethod(arr, pos - 1);
            if (arr[pos] == 'X') return temp - 1;
            if (arr[pos] == 'Y') return temp + 2;
            return temp;
        }
    }

    public static void main(String[] args) {
        char s[] = {'X', 'Y', 'X', 'X', 'Y', 'X', 'X', 'Y', 'X', 'X'};
        System.out.println(someMethod(s, s.length - 1));
    }
}
```

Program output: _____

Describe what the program does (in conversational English):

3. For each question below, circle the best answer. (10pts)

1. Which algorithm has a run-time efficiency of $O(n)$?
 - (a) Selection sort
 - (b) Binary Search
 - (c) Sequential Search /linear search
 - (d) Merge Sort
2. What is the run-time complexity of the method `public int sum(int a, int b)` that returns the sum of its inputs?
 - (a) $O(1)$
 - (b) $O(n \log n)$
 - (c) $O(\log n)$
 - (d) $O(n)$
3. In the average case, which algorithm has a run-time efficiency of $O(n \cdot \log_2(n))$?
 - (a) Selection Sort
 - (b) Binary Search
 - (c) Sequential Search
 - (d) Merge Sort
4. Which of these running times is the best (i.e. the shortest time)?
 - (a) $O(n^3)$
 - (b) $O(2^n)$
 - (c) $O(n \log n)$
 - (d) $O(\log n)$
5. What is the Big-Oh of this function: $T(n) = n^2 + 3n^3 + 12n + 5$
 - (a) $O(n^3)$
 - (b) $O(n^2)$
 - (c) $O(n)$
 - (d) $O(1)$

4. Answer the questions below regarding the complexity of the following algorithm (12pts):

```
1  public void bigoh(int n) {
2      int i, j, sum;
3      i = 0;
4      sum = 0;
5      while (i < n) {
6          System.out.println(i);
7          j = 0;
8          while (j < n) {
9              sum++;
10             j++;
11         }
12         System.out.println(sum);
13         i++;
14     }
15     System.out.println(sum);
16 }
```

- a) Count the number of times each of the following statements in the table is executed with respect to n :

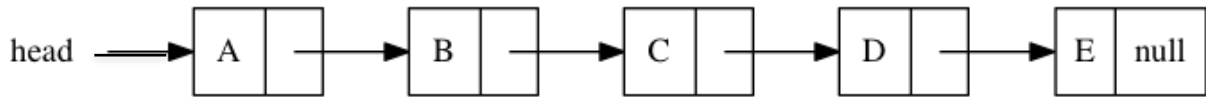
Line	# of Times Executed
3	
4	
5	
6	
7	
8	
9	
10	
12	
13	
15	

- b) The running time function, $T(n)$, of an algorithm is computed as the sum of the counts in the table above. Compute $T(n)$.

$T(n) =$

- c) What is the Big-Oh of this program?

5. LinkedList class shown below implements a singly linked list of elements of type String. You are required to first write a size() method. Then, using that method, write a method called deleteMiddle() that deletes the middle/center element from the linked list only if the list contains an odd number of elements (do nothing if the list is even in length). For example, given the list below, the node with element labeled “C” would be deleted (or removed from the list) (10pts).



If the list contains only one element, then delete that element. After deleting the middle element, make sure to adjust head instance variables as appropriate. The Node API is shown on last pages of this exam. **(Hint: First, determine the size of the list by using a loop. Use modulus -%- to determine if the size is odd).**

```
public class LinkedList {  
    private Node head; // points to head/first node in the list  
  
    public int size(){ // returns the # of elements in the list
```

```
}
```

(Proceed to the next page for deleteMiddle method)

```
public void deleteMiddle() {
```

```
    } // end deleteMiddle() method  
} // end LinkedList class
```

Node API:

`Node()` - A constructor that sets this Node's data String and next Node instance variables to null

`Node(String data, Node next)` - A constructor that sets this Node's data and next instance variables to the specified values

`String getData()` - A getter that returns the data String contained in this Node

`void setData(String data)` - A setter to set the data to the specified String

`Node getNext()` - A getter that returns the next Node that this Node points to

`void setNext(Node next)` - A setter that sets the next Node to point to the specified Node