

Ahmadu Bello University, Zaria

Department of Computer Science

COSC 301: Data Structures and Algorithm Complexity

2018/2019 First Semester Test I

Date: May 2nd, 2019

Time Allowed: 1 hour

Instructions: Attempt All questions

Student's Registration Number: U15CS1112

Signature: [Signature]

1. a) Create a visitor **SiwesVisitor** which prints the **matricno**, **accno**, and **phoneno** of each **Student** object visited only if the **accno** begins with an even number. (Hint: consider **matricno**, **accno**, and **phoneno** to be instance variables that have getters in the **Student** class with constructor **public Student(String matricno, int accno, int phoneno)**. [3 marks]

```
class SiwesVisitor extends AbstractVisitor {
    public void visit(Student obj) {
        Student stu = (Student) obj;
        if (stu.getAccno() % 2 == 0) {
            System.out.println("Matric no. " + stu.getMatricno() + ",
            Account no. " + stu.getAccno() + ", Phone no. " +
            stu.getPhoneno());
        }
    }
}
```

19
20

- b) Assume that we have a **Student** class as defined in 1a, write a test class that creates an object of **MyContainer** and then insert at least three (3) Students into the container. Then make the visitor in (1a) visits the objects in your container. [3 marks]

```
class TestSiwesVisitor {
    public static void main(String[] args) {
        MyContainer c = new MyContainer();
        c.insert(new Student("U18AA1234", 12345678, 08100123466));
        c.insert(new Student("U17ZZ4321", 87654321, 08012345678));
        c.insert(new Student("U16BD2222", 44444444, 09099999999));
        c.accept(new SiwesVisitor());
    }
}
```

- c) Compute the **exact** and **approximate** complexity of the following algorithms (show your workings): [4 marks]

Algorithm	Complexity
<pre>for (int i=1; i<= n; i=2*i){ for (int j=1; j< i; j++){ int y=auxi(j); System.out.println(y); } } public int auxi(int k){ int sum=0; for (int j=1; j< k; j++){ sum+=j; } return sum; }</pre>	<p>Exact complexity <u>outer loop</u> Initialization = 1, no. of iterations = $\log_2(n)$ Condition = $\log_2 n + 2$, index update = $\log_2 n + 1$ <u>inner loop</u> Initialization = 1, iterations = $i-1$, condition = 1 index update = $i-1$ <u>method call</u> Initialization = 1 condition = k index update = $k-1$ Approximate complexity $O(\log n + 1) * O(n-1) * O(k-1)$ $= 3 + 2\log n + [2^i + (3 + 3k)(n-1)] \log n + 1$ $= 3 + 2\log n + 5\log n + 3k\log n + 3\log n - 3k\log n$ $= 3 + 2\log n + 5\log n - 3k\log n$ $= 3 + 7\log n - 3k\log n$ $= O(k \log n)$</p>

2. a) `mlist` is a `SinglyLinkedList` object initialized with integer objects having the following values:

`mlist = {11, 20, 13, 7, 21, 10, 5, 3}`

What is the output of the call `mlist.Question4()`, if the method `Question4` defined below is assumed to be in the `MyLinkedList` class (an implementation of a `SinglyLinkedList`)? [5 marks]

```
public double Question4() {
    int c=0;
    int s=0;
    double a=0;
    Element e=head;
    while (e != null) {
        int d=((Integer)e.data).intValue();
        if (d % 2==1) {
            s+=d;
            c++;
            System.out.println(s+" "+c);
        }
        e=e.next;
    }
    a=s/c;
    return a;
}
```

Program Output...

11 1
24 2
31 3
52 4
57 5
60 6

3

What is the method computing?

The mean of the odd numbers in the list

2

b) Write a test class `Reverser` that accepts an input string and uses a stack to reverse the input string. For example, given the input, "ALPHABET", the string "TEBAHPLA" will be printed on the screen. [3 marks]

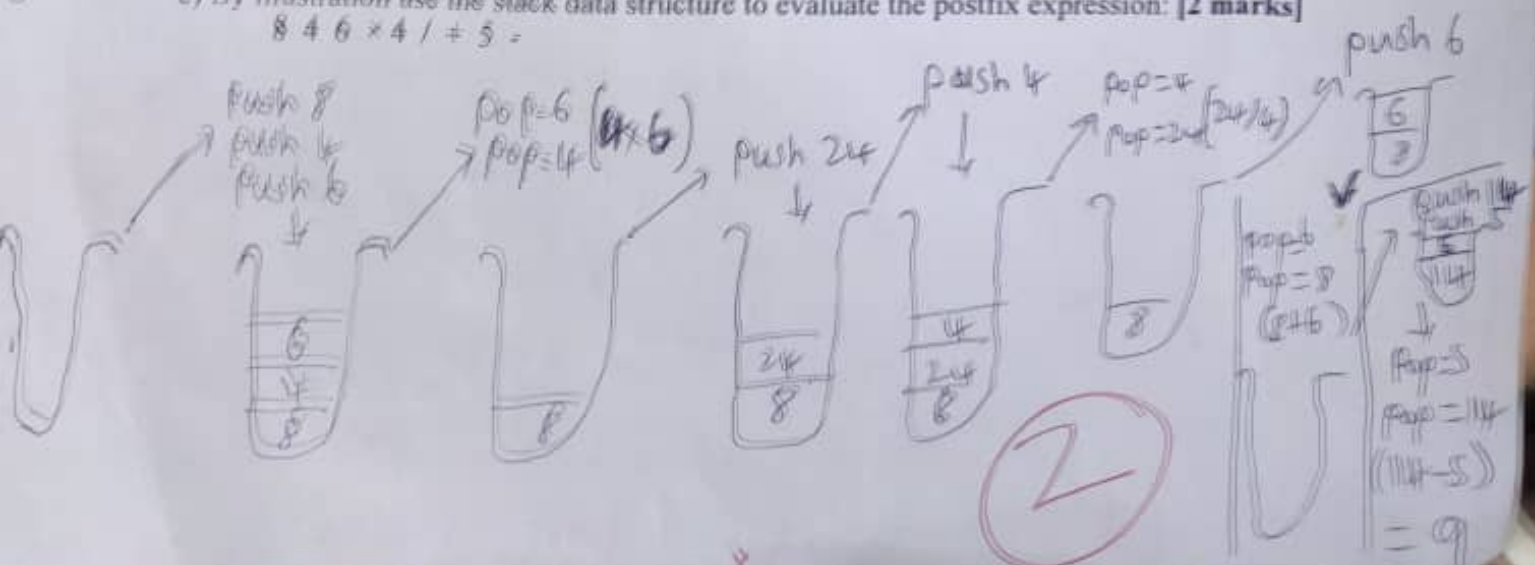
```
class Reverser {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String input = in.next();
        Stack<Character> stack = new Stack<Character>();
        for (int i=0; i<input.length(); i++) {
            stack.push(input.charAt(i));
        }
        while (!stack.isEmpty()) {
            System.out.print(stack.pop());
        }
    }
}
```

Continuation

$$= 3i^0 \log n + 5i^0 \log n - 3i^0 \log n - \log n + 3i^0 \log n + 5i^0 - 3i^0 //$$

c) By illustration use the stack data structure to evaluate the postfix expression: [2 marks]

`8 4 6 * 4 / + 5 -`



500 LEVEL

(65)

Ahmadu Bello University, Zaria
Department of Computer Science

2017/2018 First Semester Examination
COSC 301: Data Structures

Date: May 22, 2018

Time Allowed: 120 Minutes

Instructions:

1. Attempt ANY FOUR questions.
2. Write all your answers in the spaces provided on this Question Paper.

Invigilator's Signature:

Registration Number: **Signature:**

Date : **Time:**

Scores:

Question	Maximum Scores	Scores Obtained
1	15	
2	15	
3	15	
4	15	
5	15	
6	15	
Total	60	

FILL IN THE QUESTIONS YOU ANSWERED IN THE BOXES BELOW

Questions Answered:				
---------------------	--	--	--	--

1. a) (5 Marks) Consider that we have class **Staff** with constructor **Staff (int staffID, String name)** and this class has its getters and setters. Create a visitor **StaffSearcherVisitor** that checks for each object (staff) visited if its name begins with 'A' then prints staff name concatenated with "Eligible". But when the object (staff) name does not begin with 'A', it prints staff name concatenated with "Not Eligible".

b) (5 Marks) Write a test class called **TestMyStaffSearcher** that creates an object of **MyContainer** and then insert into the container five (5) staff objects. Then make your container to use the visitor you created in question (1a) above. Also, use an object of **Iterator** to print out all the items in the container.

c) (5 Marks) Assume we have class **Staff** with constructor **Staff (int staffID, String name)**. Write a test class **TestAssociation** that creates an object of **MySearchableContainer** known as the staff container, and then insert four staff into it. Also, create an object of **Department** with constructor **Department (String name)**, then associate, using an **Association** object, the department to the staff container before inserting the object of the association into another object of **MySearchableContainer**, known as association container.

2. a) (4 Marks) Given a collection of algorithms that runs on

$O(1)$, $O(n \log n)$, $O(n)$, $O(n^2)$, $O(\log n)$ and $O(n!)$

order the algorithms from fastest to slowest.

--

b) (5 Marks) Determine the number of basic operations performed by the method Question2(). Show details of your derivations.

<pre>static int Question2 (int n){ int sum=0; for (int i=1; i<n; i=i*2) sum=sum+i+ find(i); return sum; } static int find (int n){ int sum=0; for (int i=1; i<=n; i++) sum=sum + i; return sum; }</pre>	
--	--

c) (6 Marks) Given that n is the problem size, determine the Big-O complexity of the program fragment given below.

S/N	Program Fragment	Complexity
i.	<pre> int sum = 0; if(x > 12){ for(int i = 0; i < n; i++){ for(int k = 0; k < n * n; k++){ sum++; } } } else { for(int i = 1; i < 10000; i = i * 2) sum++; } </pre>	
ii.	<pre> for (int i = 0; i < n; i++) for (int k = 0; k < i; k++) sum++; </pre>	

3. a) (3 Marks) Write a method that returns a linked list whose objects have been reversed so that the first element is now last and the last now first. For example, a list of the form [4,5,7,9,2] should be [2,9,7,5,4]. This method should have this signature: **public MyLinkedListReverser()**.

b) (5 Marks) Create a class **TestMyLinkedListReversal** which creates an object of **MyLinkedList** and populates it from the end with values 0-9, and print out its items. Then call the method **listReverser()** and then display the content of the returned linked list.

c) (7 Marks) Create a class `TestMyQueueSeparator` which creates an object `allQueue` of `QueueAsLinkedList` and then prompts user to enter ten (10) numeric values which are add to the queue object. Also, create two (2) new objects of `QueueAsLinkedList` named `oddQueue` and `evenQueue`. Then, iterate over `allQueue`, if an item is prime add to the `evenQueue` otherwise to `oddQueue`. Finally, print the content of the three queues.

starting from the n...
 21. in 1938, ending up with the IBM701.
 The technological development of computers.

4. a) (6 Marks) Evaluate the postfix expression:
 $5, 20, 15, -, *, 25, 2, *, +$

by completing the table below:

Step	Input Symbol	Action	Stack Status	Intermediate Output
1	5	push	5	-
2	20	push	5, 20	-

- b) (5 Marks) Complete the recursive method `powerN(int base, int n)` below such that:
 $\text{powerN}(3,0) = 3$ (i.e. 3^0)
 $\text{powerN}(3,1) = 3$ (i.e. 3^1)
 $\text{powerN}(3,2) = 9$ (i.e. 3^2)

```
public int powerN(int base, int n){
```

```
}
```

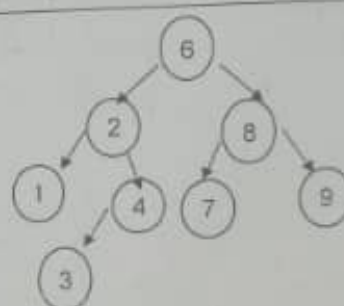
- c) (4 Marks) Trace the method call: `powerN(3,4)`. State the number of activation records.

3001

5. a) (4 Marks) Write a method which returns the number of nodes in a binary tree. The method header is given below:
- ```
public int numOfNodes(BinaryTree t){
```

}

- b) (1 Mark each) Given the tree here, what is the:
- (i) the size
  - (ii) Height
  - (iii) Preorder traversal output
  - (iv) Postorder traversal output
  - (v) Inorder traversal output.



- c) (6 Marks) Insert the keys 69, 47, 75, 30, 23, 81, and 79 in this order in an initially empty B-tree of order 3

# THE OFF

6. a) (10 Marks) Starting with an empty tree, construct an AVL tree by inserting the following keys in the order given: 2, 3, 5, 6, 9, 8, 7, 4, 1. If an insertion causes the tree to become unbalanced, then perform the necessary rotations to maintain the balance. State where the rotation were done and also what type of rotation.

Given that the following array represents a heap stored in an array with index starting from 1.

| 1  | 2  | 3  | 4  | 5  | 6  |
|----|----|----|----|----|----|
| 95 | 77 | 88 | 11 | 45 | 85 |

Use the array to answer question 6 (b) and 6 (c).

b) (2 Marks) Show the tree representation of the array

c) (3 Marks) Show the content of the array after 105 is inserted into the original heap.

300 LEVEL PAST QUEST

OFFICE FLAT FILE

Ahmadu Bello University, Zaria  
Department of Mathematics

2015/2016 First Semester Examination  
COSC301: Data Structures

Date: April 11, 2016

Time Allowed: 120 Minutes

Instructions:

1. Attempt ANY FOUR questions.
2. Write all your answers in the spaces provided on this Question Paper.

Student's Registration Number: \_\_\_\_\_ Signature: \_\_\_\_\_

Date of Examination: \_\_\_\_\_ Time: \_\_\_\_\_

Scores:

| Question | Maximum Scores | Scores Obtained |
|----------|----------------|-----------------|
| 1        | 20             |                 |
| 2        | 20             |                 |
| 3        | 20             |                 |
| 4        | 20             |                 |
| 5        | 20             |                 |
| 6        | 20             |                 |
| Total    | 80             |                 |

Questions Attempted: \_\_\_\_\_

# 300 LEVEL PAST QUEST

(20 marks). Select the most appropriate matching answer for each of the following questions. Each question is worth two marks. Write your answers in the following table.

| i | ii | iii | iv | v | vi | vii | viii | ix | x |
|---|----|-----|----|---|----|-----|------|----|---|
|   |    |     |    |   |    |     |      |    |   |

- Let  $v$  be an arbitrary node in an arbitrary tree  $t$ . Then,  $\text{depth}(v) + \text{height}(v) = \text{height}(t)$ .  
(a) true (b) false
- Let  $t$  be a heap tree of height 3. Then, the minimum number of nodes  $t$  can have is  
(a) 0 (b) 4 (c) 8 (d) 15
- The maximum number of nodes  $t$  in the preceding question can have is  
(a) 0 (b) 4 (c) 8 (d) 15
- There exists a full binary tree of with 55 nodes.  
(a) true (b) false
- Consider a heap node  $v$  stored at  $A[j]$ . Assuming that the heap is indexed starting from 0, then if  $v$  has a right child, it will be stored at  
(a)  $A[2j]$  (b)  $A[2j+1]$  (c)  $A[2j-1]$  (d)  $A[2j+2]$
- Based on the way we defined equality on Association objects in our code, what is the result of `a.equals(b)` for the following pair of Associations objects?  
  

```
Comparable a=new Association (new Integer(3), new Integer(4));
Comparable b=new Association (new Integer(3));
```

  
(a) true (b) false
- How many times does the following loop iterate? `for(int i=k; i<n; i=i*m) loopBody;`  
(a)  $\log_m n$  (b)  $\log_2 n$  (c)  $\lceil \log_m \frac{n}{k} \rceil$  (d)  $\lceil \log_m \frac{n}{k} + 1 \rceil$
- Write down the result of evaluating the postfix expression: `3 2 * 6 4 + 2 * -`.  
(a) 20 (b) -20 (c) 14 (d) -14
- To what complexity class does the following loop belong?  

```
for(int i=1, sum=0; i<100000; i=i*2)
 sum++;
```

  
(a)  $O(\log n)$  (b)  $O(n)$  (c)  $O(100000)$  (d)  $O(1)$  (e) none of these

- xi) What is the result of decoding the bits: 1001000100011011000110 using the following Huffman codewords?

| Character | A  | I  | M  | N  |
|-----------|----|----|----|----|
| Codeword  | 10 | 00 | 11 | 01 |

- (a) ANININAMINA (b) ANINIAMINA (c) ANINIANMINA (d) none of these



## 300 LEVEL PAST QUEST

20 marks). Consider the Room class:

```
class Room{
 private int roomID;
 private int occupants;
 private String hallName;

 public int getRoomID(){
 return roomID;
 }
 public int getOccupants(){
 return occupants;
 }

 public String getHallName(){
 return hallName;
 }

 // ... more code left out ...
}
```

- (a) (10 marks). Create a visitor called **RoomAccommodationVisitor**. This visitor should be capable of visiting each Room to check those with occupant more than 4, and then displays information in this format: "roomID in hallName has squatters".

300 LEVEL PAST QUEST

- (b) (10 marks). Create a `TestRoomAccommodationVisitor` class that inserts at least two instances of `Room` into a container and then applies the `RoomAccommodationVisitor` on elements of the container.

300 LEVEL PAST QUEST

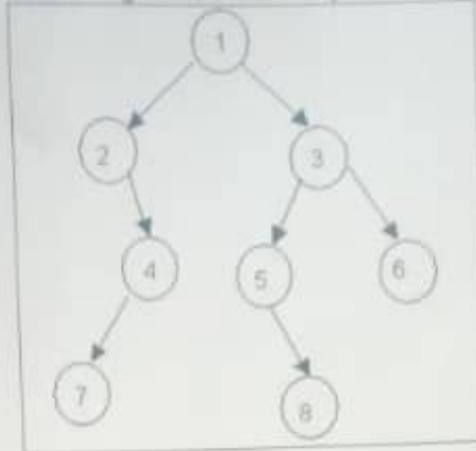
## THE OFFICE FLAT FILE

(20 marks). Answer both the following questions.

(a) (10 marks). Form and solve the recurrence relation for the running time of Fibonacci method below and hence determine its big-O complexity:

```
long fibonacci (int n) { // Recursively calculates Fibonacci number
 if (n == 1 || n == 2)
 return 1;
 else
 return fibonacci(n - 1) + fibonacci(n - 2);
}
```

(b) (10 marks). Use the following tree to answer the questions below.



- i. List nodes of this tree in preorder traversal sequence.
- ii. List nodes of this tree in inorder traversal sequence.
- iii. List nodes of this tree in postorder traversal sequence.
- iv. Is this tree a binary heap? If not, explain why.

4. (20 marks). Answer both the following questions.

- a. (10 marks). Implement a `TestStudentQueue` that creates an object of `QueueAsLinkedList`. Insert at least two students into the `QueueAsLinkedList`. The `Student` class constructor is of the form `Student(int ID, boolean isUG)` and has a `getID()`. Use the `Iterator` class to iterate each `Student` and then uses a method of `isUGStudent()` of the `Student` class which returns true when the student is an undergraduate and false otherwise. Your `Iterator` should print only undergraduate students.

- b. (10 marks). Study the following program carefully and answer the questions that follow.

```
void recursive(int n){
 if(n>1) {
 recursive(n-1);
 System.out.println(n);
 recursive(n-1);
 }
}
```

- i. (5 marks). How many lines of output are produced, if any, by the call `recursive(5);`?
- ii. (5 marks). Write down the last five lines of output, if any, for the call `recursive(5);`.



# 300 LEVEL PAST QUESTIONS THE OFFICE FILE

(10 Marks). Answer both the following questions.

- a. (10 marks). Insert the keys: 11, 1, 6, 3, 5, 16, 12, 15, and 13, in this order, into an initially empty AVL tree. Whenever a rotation is needed, write the kind of rotation you make and the key that causes the rotation in the following table. Just write SL (for Single Left rotation), SR (for Single Right rotation), DLR (for Double Left-Right rotation) and DRL (for Double Right-Left rotation) as appropriate in each stage.

- b. (10 Marks). Write a method in Java that takes an array representing a binary heap as parameter and prints each node that has a right child in the heap. Assume the root node of the heap is stored at index 1 in the array.

# 300 LEVEL PAST QUESTIONS THE OFFICE FLAT FILE

(20 marks). Answer both the following questions.

- a. (10 marks). Draw the final MaxHeap obtained when the following sequence of numbers is used to build a MaxHeap, bottom-up:

9 8 13 15 16 18 12 35 22 31

- b. (5 marks). Draw the resulting trees in deleting 30 and 5 from the B-tree of order 5 given below. Deletion of 5 should be on the B-Tree obtained from deleting 30.

