UNIVERSITY
OF
JOHANNESBURG

FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC03A3/CSC3A10 COMPUTER SCIENCE 3A
CAMPUS	AUCKLAND PARK CAMPUS (APK)
EXAM	JUNE EXAM 2019

DATE: 2019-05-25	**SESSION:** 12:30 - 15:30
ASSESOR(S):	PROF D.T. VAN DER HAAR MR R. MALULEKA
EXTERNAL MODERATOR:	PROF J. GELDENHUYS (SUN)
DURATION: 180 MINUTES	**MARKS:** 150

Please read the following instructions carefully:

1. Answer **all** the questions
2. Write *cleanly* and *legibly*.
3. You may use a non-programmable calculator to answer the questions.
4. This paper consists of 7 pages.

QUESTION 1

(a) Analyze the code below (which performs a division operation on two integers) and answer the questions that follow. (5)

```
public int performCalc(int a, int b){
try{
   if (b==0)
   return a/b;
else
   return -1.0;
}
catch (NumberFormatException e){
   System.err.println("An error occured");
}
}
```

1. Is there anything **wrong** with the code segment?
2. Justify your answer.

(b) Within the body of a method in Java, the keyword **this** is automatically defined as a reference to the instance upon which the method was called. Discuss two reasons why the **this** reference could be needed in a method body. (4)

(c) How would you go about inserting an element at the **tail** of a **doubly linked list**? You may use pseudo code or diagrams to support your answer. (6)

Total: 15

QUESTION 2

(a) Consider the following function and using **primitive counting**, express the runtime of this function in Big-Oh notation, along with a justification for your answer. (5)

```
public int[] prefAvg(int[][] X, int n){
  int s = 0;
  for(int i = 0; i< n;i++){
    for(int j = 0; j< n;j++){
      if (X[i][j]>5)
        s++;
    }
  }
  return s;
}
```

(b) Provide Java source that represents an **iterative** version of the following function. (5)

```
public static int factorial(int n) throws
    IllegalArgumentException {
    if (n < 0)
        throw new IllegalArgumentException( );
    else if (n == 0)
        return 1;
    else
        return n * factorial(n-1);
}
```

(c) An obvious choice for implementing the **ArrayList ADT** is to use an extendable array. Name **two strategies** for providing the ability to expand the array, together with a brief amortized analysis for a **single add operation** for each strategy. Which of the two strategies would you implement if you wanted to save on total running time? (5)

Total: 15

QUESTION 3

(a) Consider the following List Interface and write a class *Queue* that makes use of the List Interface and the Adapter design pattern to realize a *Queue ADT*. **Note: You do not need to implement the List methods.** (10)

```
    public interface List<T> {
    public Node<T> addAfter(Node<T> elem, T item);
    public Node<T> addFirst(T item);
    public Node<T> addLast(T item);
    public T remove(Node<T> elem);
    public Node<T> search(T elem);
    public Node<T> first();
    public boolean isEmpty();
    public Integer size();
    }
```

(b) Discuss how a Priority Queue can be used to sort a set of comparable elements. Be sure to include the two possible implementations together with the performance of the methods for these implementations. (5)

Total: 15

QUESTION 4

(a) What is the maximum number of nodes n in a binary tree of height h? Show all your calculations. You may use a diagram to support your answer. (5)

(b) Illustrate the execution of the **bottom-up construction of a heap** on the following sequence. You only need to provide a graphic representation of the heap at each stage in the construction, including any intermediate operations. (10)

(15, 4, 10, 20, 13, 25, 6, 18, 30, 17, 40, 12, 29, 8, 31)

Total: 15

QUESTION 5

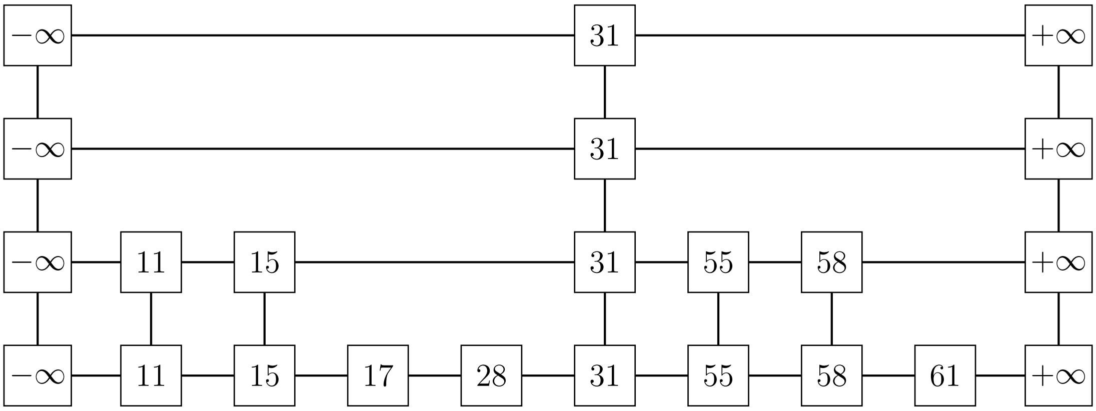
(a) What is the worst case performance (in Big Oh notation) of the *put* and *get* operations in a **List-based Map**. (2)

(b) Given a hash function $h(x) = xmod13$ for a hash table that uses **linear probing**, redraw the hash table below and **insert** the keys 18, 41, 22, 44, 59, 32, 31, 73 in this order. (8)

0	1	2	3	4	5	6	7	8	9	10	11	12	13

(c) Aside from linear probing-based collision handling, name **and** describe two (2) **other collision handling methods** that can be used in a hash table. (4)

(d) Analyse the skip list below and illustrate using diagrams how you would **insert** an entry with a key of 56 and 3 heads coin flips. (6)



Total: 20

QUESTION 6

Database management systems often use **tree-based structure** to maintain indices for a particular table. Compare and contrast **three** tree-based structures discussed in class, discuss their **advantages** and **disadvantages**, as well as the **run-time efficiencies** for each tree structure discussed. Discuss their **applicability** for use in a database management system. Furthermore, indicate **which** tree-based structure would you choose as the best structure for this purpose, clearly indicating the reasons for your choice.

Total: 15

QUESTION 7

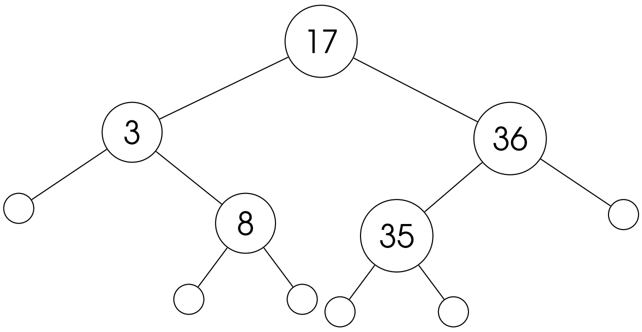
Consider the following AVL tree provided below. Draw the AVL tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced. Removal operations should follow from the tree that resulted from the insertion operations.

1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order)

   46, 29, 2, 24, 39, 6, 4

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order)

   3, 4, 6, 24



Total: 15

QUESTION 8

Consider the following Red-Black tree provided below. Draw the Red-Black tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced. Removal operations should follow from the tree that resulted from the insertion operations.Removal operations should follow from the tree that resulted from the insertion operations.

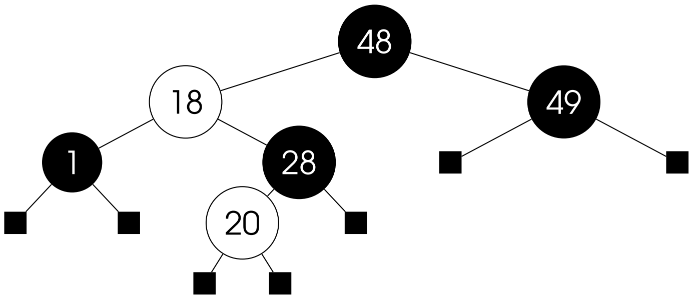
1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order)

   16, 22, 16, 37

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order)

   16, 49, 20, 28, 18, 22, 1, 48

The Red-Black tree is in the current state:



Total: 20

QUESTION 9

(a) Define a **(2,4) tree**, along with **properties** it portrays and provide a **diagram** that demonstrates its structure. (5)

(b) Bob loves programming languages and wants to plan his university course schedule for the next few years (assuming that he is allowed to choose). He is interested in the following nine programming language courses: ASM, C, C++, Java, C#, Erlang, Scala, SML and Smalltalk. The prerequisites for the courses are: (10)

- ASM - none
- C - ASM
- C++ - ASM, C
- Java - C, C++
- C# - Java, ASM, C++
- Erlang - SML, Smalltalk
- Scala - Java, SML, Smalltalk
- SML - ASM
- Smalltalk - none

Draw a **graph** that demonstrates the dependencies between the above courses, and then outline the order of courses Bob should take to cover all the languages he wants to learn.

(c) Provide the steps for a **Depth First Search** graph traversal. (5)

Total: 20

— End of paper —