FACULTY OF SCIENCE
ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC03A3: COMPUTER SCIENCE 3A
CAMPUS	AUCKLAND PARK CAMPUS (APK)
ASSESSMENT	A

DATE:	**SESSION:**
ASSESOR(S):	PROF D.T. VAN DER HAAR MR R. MALULEKA
EXTERNAL MODERATOR:	DR. E. JEMBERE (UKZN)
DURATION: 180 MINUTES	**MARKS:** 150

Please read the following instructions carefully:

1. You must complete this assignment yourself within the prescribed time limits.
2. You are bound by all university regulations. Please take special note of those regarding assessment, plagiarism, and ethical conduct.
3. No communication concerning this test is permissible during the assessment session except with Academy staff members.
4. This paper consists of 6 pages excluding the cover page.

QUESTION 1

(a) Analyse the Java source below and answer the questions that follow: [4]

```
1  public class Animal{
2    private String name;
3    private int age;
4  }
5
6  public class Cat extends Animal {
7    private String felineCategory;
8  }
9
10 public static int main(String[] args) {
11   Cat c = new Animal();
12   System.out.println(c.name);
13 }
```

1. What is **wrong** with the above source code?
2. How would you **fix** this/these problem(s)?

(b) Write a recursive Java function that realises the following expression: [4]

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

(c) Analyze the code below (which computes array element summation) and answer the questions that follow. [7]

```
1 public int A(int[] arr, int i) {
2   return arr[i] + A(arr,i++)
3 }
```

1. Draw a recursion trace for A(a, 0) where a = {7,5,2,1}. [4]
2. Give an iterative version of the function. [3]

Total: 15

QUESTION 2

(a) Compare and contrast the implementation of a Singly-linked and Doubly-linked list. Provide two (2) points of comparison with regards to the structural differences between these structures, and discuss three (3) advantages and disadvantages of these structures. [5]

(b) What is the asymptotic relationship between each of the following pairs of functions? [4]

1. $3n^3 + n - 5$ and $n^3 + 4n + 7$
2. 2^n and 1^n

(c) Consider the following function and use primitive counting to determine its runtime as a function of n, the length of the array. Show all calculations and be sure to provide the
final answer in Big-Oh notation. [6]

```java
int b(int arr[], int x)
{
  int l = 0;
  r = arr.length - 1;
  while (l <= r) {
    int m = l + (r - l) / 2;
    if (arr[m] == x)
      return m;
    if (arr[m] < x)
      l = m + 1;
    else
      r = m - 1;
  }
  return -1;
}
```

Total: 15

QUESTION 3

(a) Consider a Java *Array* of generic type *T*. Write a class **Stack** that makes use of the Array to realize a **Stack ADT**. *Note - Exception handling does not have to be included.* [10]

(b) A Priority Queue can be implemented using a heap or an array. Explain the two (2) possible implementations when using an **array** to implement a Priority Queue and the impact that these implementations have on the **runtime** of the Priority Queue structure. What would the runtime of **heap**-based Priority Queue be for a removal operation? [5]

Total: 15

QUESTION 4

(a) Define the concept of **depth** and discuss how depth in a binary tree is calculated recursively. [4]

(b) Explain why a Binary Tree that is not a Complete Binary Tree cannot have an efficiency equivalent to $O(\log n)$. [5]

(c) 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. [6]

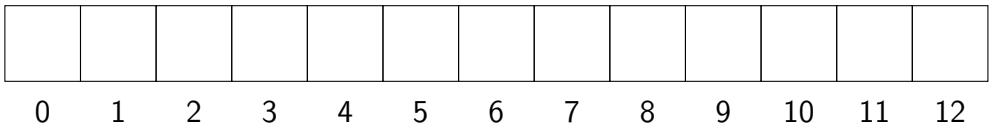
(16, 3, 33, 48, 29, 11, 5)

Total: 15

QUESTION 5

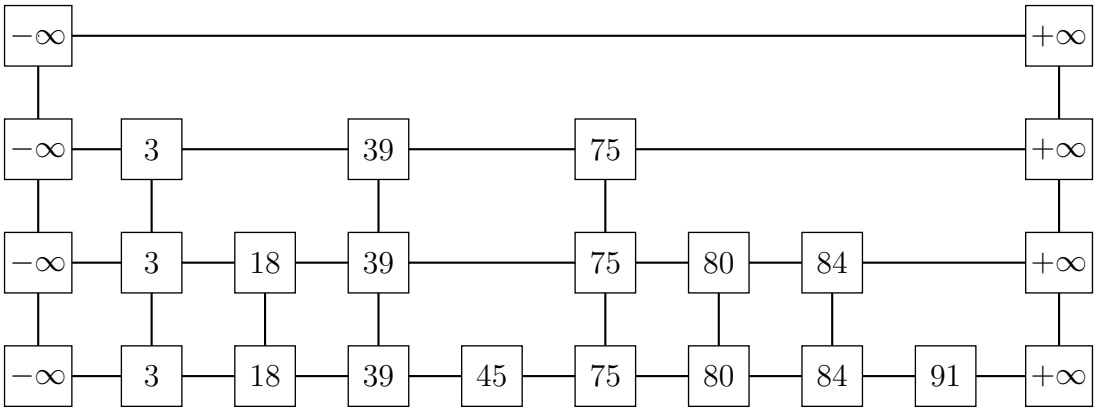
(a) How does the performance of a **list-based map** compare to that of a **hash-table map** implementation that uses separate chaining to handle collisions? For each implementation, give the performance of each method - *get*, *put*, and *remove* - in Big Oh notation. [6]

(b) Given a hash function $h(x) = xmod11$ for a hash table that uses **linear probing**, redraw the hash table below and **insert** the keys 13, 20, 8, 90, 32, 6, 18, 30 in this order. [8]



(c) What is the **load factor** for the above hash table **after** all the entries have been inserted? [2]

(d) Analyse the skip list below and illustrate using diagrams how you would **insert** an entry with a key of 60 and 2 heads coin flip. [4]



Total: 20

QUESTION 6

(a) You have been contacted by the *Police Service* to work on their new national DNA database. Your system will be used to store the DNA profiles of known offenders, and to allow police to match samples found at crime scenes against stored records. Your system must be able to efficiently handle frequent insertions and searches. [15]

Discuss **three** abstract data-types suitable for your database system. Discuss their **advantages** and **disadvantages**, as well as the **run-time efficiencies**. Indicate which data structure you choose as the best for this purpose, clearly indicating the reasons for your choice.

Total: 15

QUESTION 7

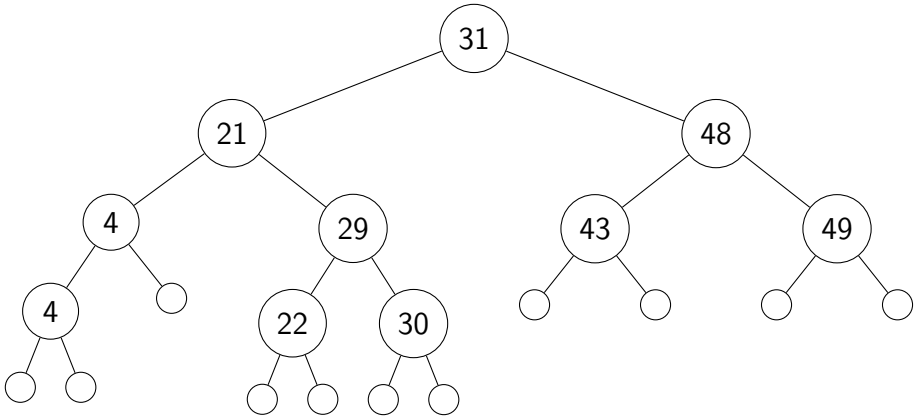
Consider the AVL tree provided below. Draw the AVL tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced. If there is a duplicate key, the inorder predecessor should be used. Removal operations should follow from the tree that resulted from the insertion operations (i.e. removals take place after all the inserts have been completed).

1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order, using an inorder predecessor duplicate strategy)

   43, 27, 3, 13, 39

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

   13, 4, 43, 4



Total: 15

QUESTION 8

Consider the (2,4) tree provided below. Draw the (2,4) tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced.

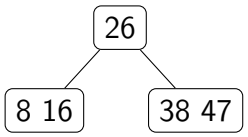
1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order, using an inorder predecessor duplicate strategy)

   37, 0, 15, 11, 40, 20, 47

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order, using an inorder predecessor removal strategy)

   47, 16, 11, 26, 38

The (2,4) tree is in the current state (the leaf nodes are not shown, however they are assumed to exist):



Total: 15

QUESTION 9

Consider the 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. If there is a duplicate key, the inorder predecessor should be used. Removal operations should follow from the tree that resulted from the insertion operations.

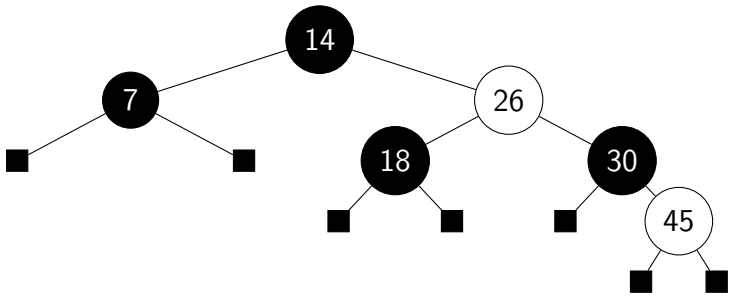
1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order)

   47, 5, 2, 13

2. Delete nodes that contain the following keys: (removed one-by-one, in the given order, using an inorder predecessor removal strategy)

   7, 18, 47, 2, 30

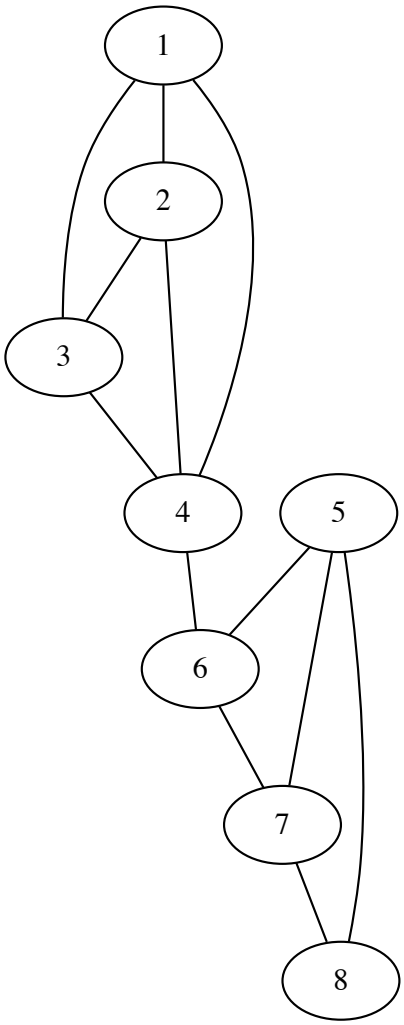
The Red-Black tree is in the current state:



Total: 15

QUESTION 10

(a) Analyse the undirected graph representation below and answer the question that follows: [10]



Show how the vertices will be visited if a **Depth First Search (DFS)** is performed, starting at **1**, along with whether you think a DFS or breadth first search (BFS) will reach vertix **8** faster. You may use a figure to support your answer.

Total: 10

— End of paper —