FACULTY OF SCIENCE

ACADEMY OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

MODULE	CSC03A3/CSC3A10 COMPUTER SCIENCE 3A
CAMPUS	AUCKLAND PARK CAMPUS (APK)
ASSESSMENT	SEMESTER TEST 2

DATE: 2022-05-19	**SESSION:** 14:00 - 16:00
ASSESOR(S):	PROF D.T. VAN DER HAAR MR R. MALULEKA
INTERNAL MODERATOR:	PROF D.A. COULTER
DURATION: 120 MINUTES	**MARKS:** 100

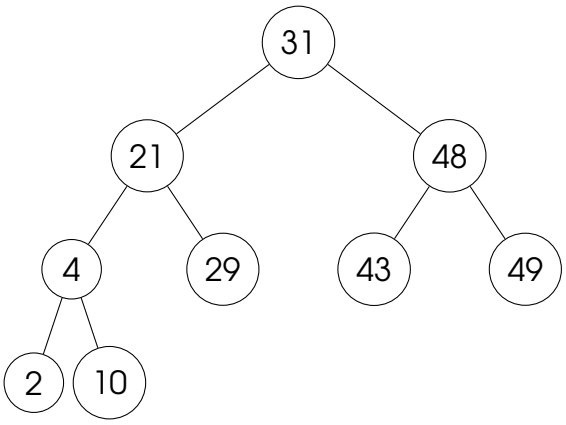
Please read the following instructions carefully:

1. You are bound by all university regulations. Please take special note of those regarding assessment, plagiarism, and ethical conduct.
2. Answer **all** the questions
3. Write *cleanly* and *legibly*.
4. You may use a non-programmable calculator to answer the questions.
5. This paper consists of 4 pages.

QUESTION 1

(a) Suppose you have a binary tree with 10 nodes. What are the minimum and maximum possible heights of the tree? You may use diagrams to aid your answer. (4)

(b) Consider the tree below, and answer the questions that follow: (16)



1. List the elements in the order of a the following traversals:
   i. PreOrder traversal (2)
   ii. PostOrder traversal (2)
   iii. InOrder traversal (2)
   iv. Euler tour (4)
2. What is the **depth** of node with element 4? (1)
3. Is the tree a **proper binary** tree? (1)
4. Where would a key of 21 be added? (2)
5. If 31 were removed, what would happen afterwards? (2)

Total: 20

QUESTION 2

(a) Discuss the **Priority Queue ADT**, along with the performance of its main operations if a **sorted sequence** is used as its underlying data structure. (7)

(b) Give the three (3) properties of a **total order relation**, within the context of the Priority Queue ADT. (3)

(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. (10)

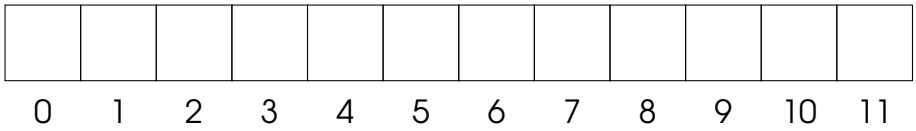
6, 9, 4, 13, 10, 55, 46, 28, 19, 12, 11, 35, 5, 38, 43

Total: 20

QUESTION 3

(a) Provide pseudo code for the **put** operation of a **List-based Map**. (5)

(b) Given a hash function $h(x) = xmod7$ for a hash table that uses **linear probing**, redraw the hash table below and **insert** the keys 6, 10, 20, 30, 27, 5, 40, 35 in this order. (8)



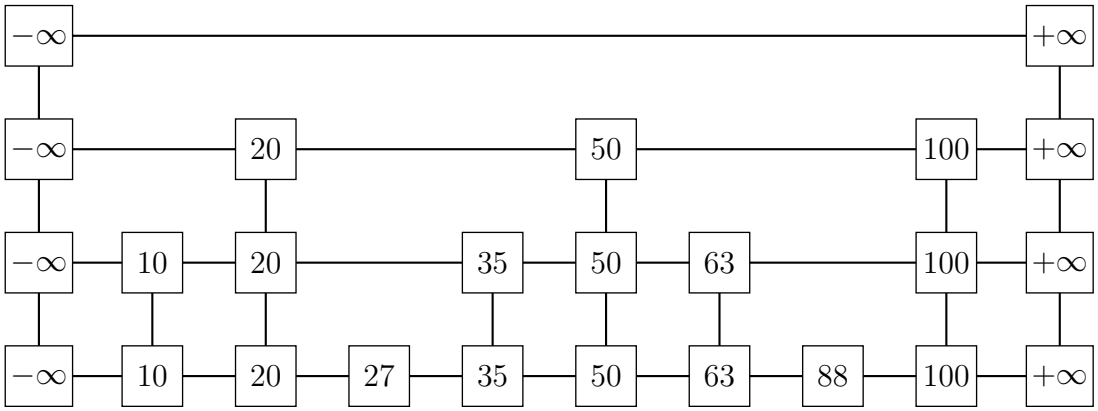
(c) What is the **load factor** for the above hash table **after** all the entries have been inserted? (2)

Total: 15

QUESTION 4

(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) Analyse the skip list below and illustrate using diagrams how you would **insert** an entry with a key of 30 and 1 heads coin flip. (6)



(c) What is the expected height of a skip list with n entries? Justify your answer. (3)

Total: 15

QUESTION 5

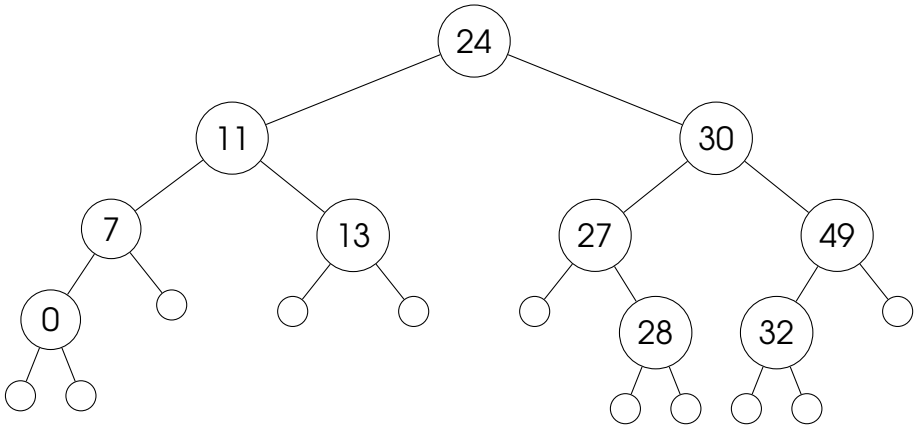
Consider the AVL tree below. Draw the AVL tree state after each of the following operations. If the tree is rebalanced draw the state before and after it being balanced. If there is a duplicate key, the inorder predecessor should be used. Removal operations should follow from the tree that resulted from 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)

   23, 5, 37, 28, 31

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

   30, 13, 31



Total: 14

QUESTION 6

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.

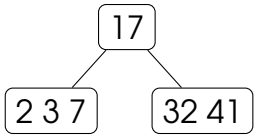
1. Insert nodes that contain the following keys: (inserted one-by-one, in the given order, using an inorder predecessor duplicate strategy)

   26, 41, 28, 40, 23, 30, 33

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

   30, 23, 2, 3, 7

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



Total: 16

— End of paper —