



**NAMIBIA UNIVERSITY  
OF SCIENCE AND TECHNOLOGY**

**FACULTY OF COMPUTING AND INFORMATICS  
DEPARTMENT OF SOFTWARE ENGINEERING**

<b>QUALIFICATION:</b> BACHELOR OF COMPUTER SCIENCE	
<b>QUALIFICATION CODE:</b> 07BCHS	<b>LEVEL:</b> 7
<b>COURSE:</b> DATA STRUCTURES AND ALGORITHMS 2	<b>COURSE CODE:</b> DSA711S
<b>DATE:</b> MAY 2025	<b>PAPER:</b> THEORY
<b>DURATION:</b> 2 HOURS	<b>MARKS:</b> 100

<b>FIRST OPPORTUNITY EXAMINATION QUESTION PAPER</b>	
<b>EXAMINER</b>	<b>PROF AMBROSE AZETA</b>
<b>MODERATOR:</b>	<b>MRS P. DOLIAN</b>

<b>INSTRUCTIONS</b>
<ol style="list-style-type: none"><li>1. Answer ALL the questions.</li><li>2. Read all the questions carefully before answering.</li><li>3. Number the answers clearly.</li></ol>

**THIS QUESTION PAPER CONSISTS OF 4 PAGES**  
(Including this front page)

- PERMISSIBLE MATERIALS**
1. NON-PROGRAMMABLE CALCULATOR

### SECTION A: TRUE OR FALSE

This section consists of 15 questions. Answer all the questions

Each correct answer is allocated 2 Marks

Write True or False for Questions 1 to 15.

- A1:** The worst-case running time for deletion in a balanced search Tree is  $O(\log n)$ . [2 Marks]
- A2:** An AVL tree is a binary search tree in which the heights of the left and right subtrees of every node differs by at most 2 [2 Marks]
- A3:** Insertion in an AVL tree starts with a regular binary search tree insertion, followed by rebalancing. [2 Marks]
- A4:** BSTs can be used to implement priority queues. [2 Marks]
- A5:** AVL tree is a self-balancing Binary Search Tree (BST) where the difference between the heights of left and right subtrees cannot be more than 1 for all the nodes. This difference is called the Balance Factor (BF). [2 Marks]
- A6:** The time complexity of KMP algorithm for pattern matching is  $O(m+n)$  [2 Marks]
- A7:** Deletion in an AVL tree starts with a regular binary search tree deletion, followed by rebalancing. [2 Marks]
- A8:** in a binary tree, every node will always have 0 or 1 children/sons. [2 Marks]
- A9:** B-tree and AVL tree have the same worst case time complexity for insertion and deletion. [2 Marks]
- A10:** In pattern matching, the output of the input statement:  
txt[] = "AAAAAAAAAAAAAAAAAAB", pat[] = "AAAAB" is: 14 [2 Marks]
- A11:** One of the methods of resolving collision in hashing is linear probing [2 Marks]
- A12:** One of the examples of probabilistic data structures is bloom filters [2 Marks]
- A13:** Operations of binary search trees does not include: Insert [2 Marks]
- A14:** The binary search algorithm finds an item on a sorted list by repeatedly discarding half of the list. [2 Marks]
- A15:**  $2-8^0$  Search Trees is a type of tree in Data Structures [2 Marks]

### SECTION B: MULTIPLE CHOICE QUESTIONS (MCQs)

This section consists of 20 MCQs. Answer all the questions

Select only one option, each correct answer is allocated 2 Marks

- B1:** The best-case time complexity for search, insertion, and deletion in an AVL tree is:  
A.  $O(\log n)$     B.  $O(n)$     C.  $\log(n)$     D.  $\log(On)$

Study Figure 1, Algorithm a and b, and use them to answer questions from B2 to B13

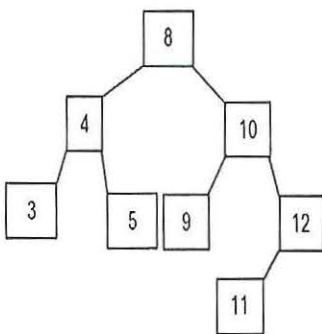


Figure 1-Binary Search Tree

```

...
public static void main(String[] args) {
    // Creating the object of Tree class
    Tree bt = new Tree();
    bt.root= new Node(100);
    bt.root.left= new Node(90);
    bt.root.right= new Node(110);
    bt.root.left.left= new Node(80);
    bt.root.left.right= new Node(95);
    bt.root.right.left= new Node(89);
}
  
```

**Algorithm a**

```

If total Nodes=3 and Node1 = Imbalance
    Right Rotate Nodes 2 and 3;
If total Nodes=3 and Node1 = Imbalance:
    For Node j = 1 to 3
        Left Rotate Node j
    End
Endif
Endif
  
```

**Algorithm b**

- B2:** In Figure 1 above, when the root node 8 is deleted, the new root node will be either:  
 A. 9 or 3      B. 4 or 10      C. 3 or 12      D. 9 or 5
- B3:** In Figure 1 above, what is the height of the tree ?  
 A. 1      B. 6      C. 3      D. 4
- B4:** In Figure 1 above, the successor of Node 12 is  
 A. 3      B. 4      C. 11      D. 9
- B5:** In Figure 1 above, is the tree a balanced BST?      Yes or No
- B6:** If node 9 is deleted, will the BST in Figure 1 be an AVL Tree?      Yes or No
- B7:** If node 3 and 5 are deleted, will the BST in Figure 1 above be an AVL Tree?      Yes or No
- B8:** Figure 1 is a valid 2-3 Tree?      Yes or No
- B9:** The height of Node 10 in Figure 1 is      A. 2      B. 0      C. 1      D. -1
- B10:** The sibling of 9 in Figure 1 is      A. 10      B. 5      C. 12      D. 11
- B11:** Postorder of the Tree in Figure 1 gives:  
 A. 8,4,3,5,10,9,12,11      B. 3,5,4,9,11,12,10,8      C. 3,4,5,8,9,10,11,12      D. 8,4,10,3,5,9,12,11
- B12:** The implementation of Algorithm a, gives:  
 A. Binary Search Tree      B. Binary Tree      C. AVL Tree      D. Namty Tree
- B13:** The AVL deletion in Algorithm b, handles:  
 A. RIGHT-LEFT Rotation      B. LEFT-Rotation      RIGHT-Rotation      D. None
- B14:** If at anytime a BST is not balanced, to restore the properties of 2-3 B Trees, re-balancing is done using  
 A. Rotation      B. Rotation and/or Exchange of colour red/black  
 C. Upward Promotion and/or Splitting      D. None
- B15:** If at any time a BST is not balanced, to restore the properties of AVL Trees, re-balancing is done using  
 A. Rotation      B. Rotation and/or Exchange of colour red/black  
 C. Upward Promotion and/or Splitting      D. None
- B16:** If at anytime a BST is not balanced, to restore the properties of Red-Black Trees, re-balancing is done using  
 A. Rotation      B. Rotation and/or Exchange of colour red/black  
 C. Upward Promotion and/or Splitting      D. None
- B17:** A function is called recursion  
 A. If it accepts a variable in high-level language      B. If it calls itself  
 C. If it translates assembly language into machine language      D. None of the mentioned
- B18:** The property that "The root and leaves (nil) are black" applies to  
 A. BST      B. Binary tree      C. AVL      D. red-black trees
- B19:** In a 2-3 tree, what is the maximum number of keys/elements a node can have?  
 A. 2      B. 1      C. 3      D. 4
- B20:** One of the following is the result of factorial (n) or n! where n = 3  
 A. 2      B. 24      C. 6      D. 120

**SECTION C: STRUCTURED/THEORY QUESTIONS**

This section consists of 3 questions. Answer **ALL** the questions  
Each correct answer is allocated 10 Marks

**Question One [10 Marks]**

- (A) Show how Hashing can be used to store the following elements: 74, 77, 65, using a Hash table of size 7. **(5 Marks)**
  - (i) Describe what happens when you want to insert a new element 44.
  - (ii) Use two methods - chaining and linear probing to resolve any collision in (i) above.
  
- (B) Using dynamic programming approach, write a function in Java that will print the first 10 numbers of Fibonacci series as follows: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 **(5 Marks)**

**Question Two [10 Marks]**

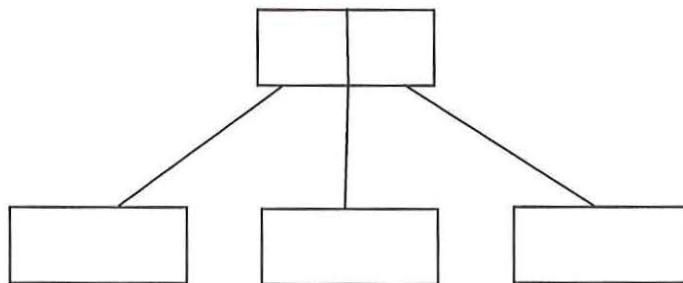
- (A) Show the steps in inserting the following elements in a Red-Black Tree: 10, 18, 7 **(5 Marks)**
- (B) Show the insertion of the following elements in a 2-3 Tree: 50 60 70 40 30 **(5 Marks)**

**Question Three [10 Marks]**

- (A) (i) In pattern matching, give the output of the below statements and explain how naive search algorithm was used to derive the output **(2 Marks)**

Input: text = "aabaacaadaabaaba",  
pattern = "aaba"

- (ii) Create by inserting the elements: 5, 7, 3, 70, 40, 83, in a 2-3 B Tree one after the another. The final tree should look like the figure below. **(3 Marks)**



- (B) (i) Briefly explain the KMP Algorithm for Pattern Matching with example(s) **(1 Mark)**
- (ii) Briefly define Dynamic Programming **(1 Mark)**
- (iii) Create and insert the elements below in a BST: 28, 22, 8 **(3 Mark)**  
Is the BST an AVL? If yes, then stop and do nothing.  
If No, then do re-balancing using ROTATION

————— THE END —————



**PAMIBIA UNIVERSITY**  
OF SCIENCE AND TECHNOLOGY

**FACULTY OF COMPUTING AND INFORMATICS**  
DEPARTMENT OF SOFTWARE ENGINEERING

<b>QUALIFICATION:</b> BACHELOR OF COMPUTER SCIENCE	
<b>QUALIFICATION CODE:</b> 07BCMS	<b>LEVEL:</b> 7
<b>COURSE:</b> DATA STRUCTURES AND ALGORITHMS 2	<b>COURSE CODE:</b> DSA711S
<b>DATE:</b> MAY 2025	<b>PAPER:</b> THEORY
<b>DURATION:</b> 2 HOURS	<b>MARKS:</b> 100

<b>FIRST OPPORTUNITY EXAMINATION QUESTION PAPER MEMORANDUM</b>	
<b>EXAMINER</b>	<b>PROF AMBROSE AZETA</b>
<b>MODERATOR:</b>	<b>MRS P. DOLIAN</b>

**THIS MEMORANDUM PAPER CONSISTS OF 10 PAGES**  
(Including this front page)

**INSTRUCTIONS TO MODERATOR/FIRST EXAMINER**

1. Please use the memorandum or sample solutions to guide your marking.
2. When marking questions you should be guided by the allocation of marks.
3. Sample answers or solutions appear in bold.
4. Reasonable, in depth or innovative correct solutions provided by the students should be allocated marks even though not provided in this memorandum
5. All things that should not be marked, e.g. any "rough work", have to be crossed out unambiguously