NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATICS
DEPARTMENT OF SOFTWARE ENGINEERING

QUALIFICATION: BACHELOR OF COMPUTER SCIENCE	
QUALIFICATION CODE: 07BCHS	**LEVEL**: 7
COURSE: DATA STRUCTURES AND ALGORITHMS 2	**COURSE CODE**: DSA711S
DATE: JULY 2025	**PAPER**: THEORY
DURATION: 2 HOURS	**MARKS**: 100

SECOND OPPORTUNITY/SUPPLEMENTARY EXAMINATION QUESTION PAPER	
EXAMINER	**PROF AMBROSE AZETA**
MODERATOR:	**MRS P. DOLIAN**

INSTRUCTIONS
1. Answer ALL the questions. 2. Read all the questions carefully before answering. 3. Number the answers clearly.

THIS QUESTION PAPER CONSISTS OF 6 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.

Study the Trees below in Figures a, b and c, and use them to answer the questions in A1, A2, A3

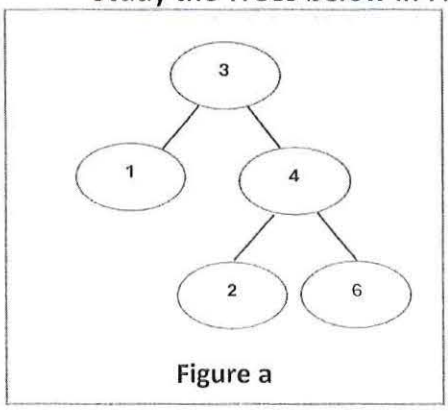


Figure a

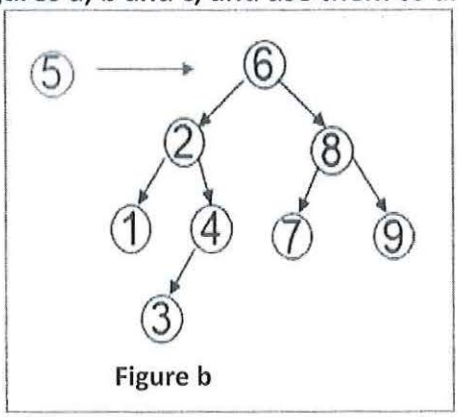


Figure b

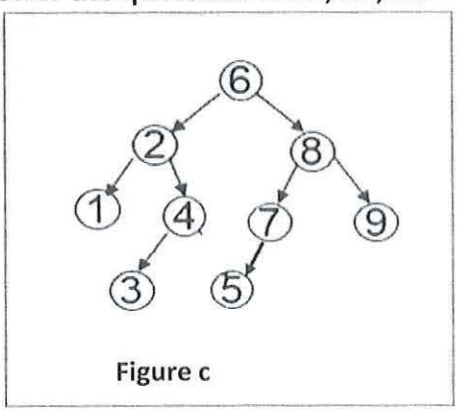


Figure c

A1: Figure **a**, is a Binary Search Tree (BST) **[2 Marks]**
A2: When 5 is inserted in BST Figure **b**, the resulting tree will not be the BST in Figure **c**. **[2 Marks]**
A3: Figure **c** is a Binary Search Tree **[2 Marks]**
A4: AVL is a type of Balanced Binary search tree **[2 Marks]**
A5: 1-6-2 Search Trees is a type of tree in Data Structures **[2 Marks]**

A6: Repeated search of half of binary tree is done in binary search tree algorithm to find a key/element. **[2 Marks]**

A7: Deletion is not one of the operations of binary search trees **[2 Marks]**
A8: Bloom filters is one of the examples of probabilistic data structures **[2 Marks]**
A9: Linear probing is one of the methods for resolving collision in hashing **[2 Marks]**

A10: The output of the input statement: *txt= "aabaacaadaabaaba", pat = "aaba"* is: [9, 0, 12] in pattern matching **[2 Marks]**

A11: Insertion and deletion have the same worst case time complexity in B-trees and AVL trees. **[2 Marks]**

A12: Every node will always have 2 or 3 children/sons in binary tree. **[2 Marks]**

A13: It is possible for a balanced BST to turn to unbalanced BST after deleting a node **[2 Marks)**
A14: The time complexity of naive pattern matching algorithm is O(m/n) **[2 Marks)**

A15: A Balance Factor (BF) is the difference between the heights of left and right subtrees **[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: A Each position of the Hash table is often called

A. Slot B. Variable C. Identifier D. HashMe

B2: One of the following defines an unbalanced BST

A. A BST is unbalanced if the difference between the height of left and right subtree for every node is more than 1.
B. A BST is unbalanced if the difference between the height of left and right subtree for every node is less than 1.
C. A BST is unbalanced if the difference between the height of left and right subtree for every node is more than 2.
D. A BST is unbalanced if the difference between the height of left and right subtree for every node is more than

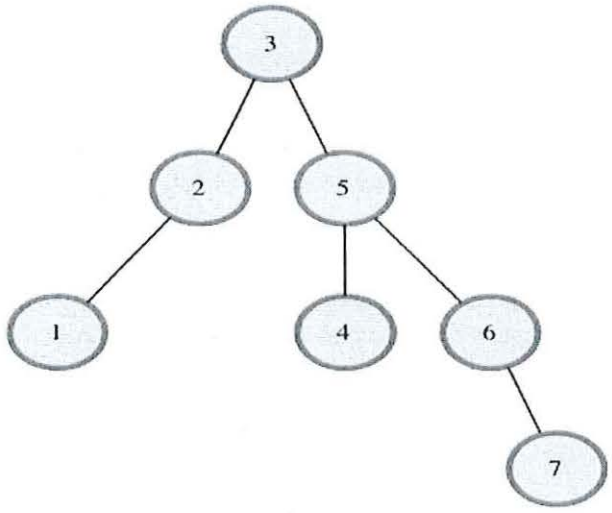


Figure 2

B3: When 7 is deleted from the above Tree in Figure 2, is the Tree a BST ? Yes or No
B4: When 7 is deleted from the above Tree in Figure 2, is the Tree AVL ? Yes or No
B5: When 7 is deleted from the above Tree in Figure 2, is the Tree Red-Black Tree ? Yes or No
B6: When 7 is deleted from the above Tree, is the Tree a 2-3 Tree ? Yes or No

B7. What will be the output of the following program?

```
main()
{
  char str[]="san foundry";
  int len = strlen(str);
  int i;
  for(i=0;i<len;i++)
     push(str[i]); // pushes an element into stack
  for(i=0;i<len;i++)
    pop(); //pops an element from the stack
}
```

A. yrdnuof nas
B. foundry nas
C. sanfoundry
D. san foundry

B8. Which of the following is an advantage of a hash table in data structures?
A. Complex to implement
B. Faster access of data
C. Exhibit low locality of reference
D. Very inefficient for less number of entries

B9. Postorder of the Tree in Figure **c** gives:
A. 6,2,14,3,8,7,5,9 B. 1,3,4,2,5,7,9,8,6 C. 1,2,3,4,6,5,7,8,9 D. 112233344455566

B10: 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

B11: Which of the following is the most widely used external memory data structure?
A. AVL tree
B. B-tree
C. Red-black tree
D. Both AVL tree and Red-black tree

B12: 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

B13: 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)

B14: One of the following is the result of factorial (n) or n! where n = 4
A. 2 B. 24 C. 6 D. 120

B15: What is a data structure?
A. A programming language
B. A collection of algorithms and codes
C. A way to store and organize data
D. A type of computer hardware

B16: In developing a dynamic programming solution, the following are/is not included in the steps:
A. Characterise the structure of an optimal solution
B. Recursively define the value of an optimal solution
C. Compute the value of an optimal solution
D. All of the above

B17: In pattern matching, the output of the statement: *Input: txt = "abcab", pat = "ab"* will give:
A. [0, 3]
B. [3, 0]
C. [1, 3]
D. [3, 1]

B18: In pattern matching, the string below would give the output:
Input:
main String: "AAAABCAAAABCBAAAABC"
pattern: "AAABC"
Output:

A. [1, 7, 15]
B. [7, 1, 14]
C. [1, 7, 14]
D. [1, 7]

B19: If the following elements are inserted into BST: 21, 11, 32
Is the tree a balanced BST? Yes or No

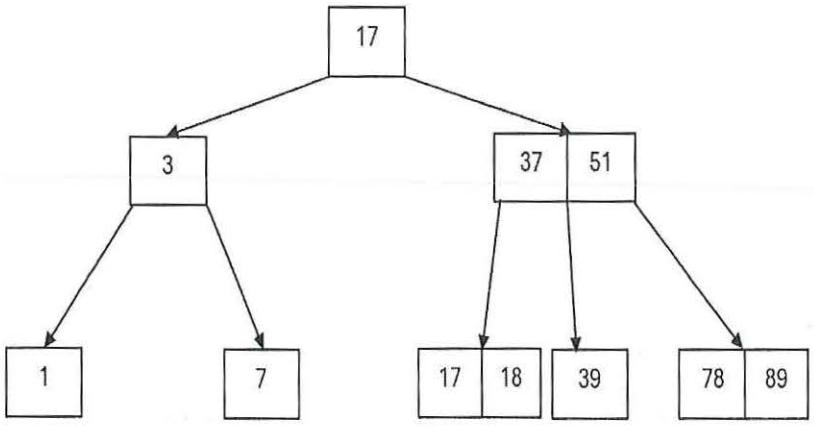
B20: If the following elements are inserted into BST: 21, 3, 11
Is the tree an AVL ? Yes or No

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) Insert 4 and 9 separately one after the order in the 2-3 B Tree below, show the new trees at every step. (5 Marks)



(B) Differentiate between KMP Algorithm for pattern matching and dynamic programming with example(s) (5 Marks)

Question Two [10 Marks]

(A) (i) Create and insert the following elements in a BST: 14, 26, 35. (5 Marks)
(ii) If the BST an AVL? If Yes, then stop and do nothing. (5 Marks)
If No, then do re-balancing using ROTATION.
Mention the ROTATION applied.

(B) Show how Hashing can be used to store the following elements: 55, 47, using a Hash table of size 5. (5 Marks)

(i) Describe what happens when you want to insert a new element 33.

(ii) Use two methods - chaining and linear probing to resolve any collision.

Question Three [10 Marks]

(A) Create and insert the following elements in a BST: 6, 40, 25 (5 Marks)
Is the BST an AVL? If Yes, then stop and do nothing.
If No then do re-balancing using ROTATION.

(B) Use insertion operation to create a 2-3 B Tree from the following elements: 6, 13, 4, 70, 40, 80 (5 Marks)

THE END