

B2.1-R4: DATA STRUCTURE THROUGH C++

NOTE:

1. There are **TWO PARTS** in this Module/Paper. **PART ONE** contains **FOUR** questions and **PART TWO** contains **FIVE** questions.
2. **PART ONE** is to be answered in the **TEAR-OFF ANSWER SHEET** only, attached to the question paper, as per the instructions contained therein. **PART ONE** is **NOT** to be answered in the answer book.
3. Maximum time allotted for **PART ONE** is **ONE HOUR**. Answer book for **PART TWO** will be supplied at the table when the answer sheet for **PART ONE** is returned. However, candidates, who complete **PART ONE** earlier than one hour, can collect the answer book for **PART TWO** immediately after handing over the answer sheet for **PART ONE**.

TOTAL TIME: 3 HOURS

TOTAL MARKS: 100
(PART ONE – 40; PART TWO – 60)

PART ONE **(Answer all the questions)**

1. **Each question below gives a multiple choice of answers. Choose the most appropriate one and enter in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)**

1.1 Identify the incorrect statement.

- A) The resources most relevant in relation to efficiency are CPU time and internal memory.
- B) An algorithm cannot have same Big-O measure for the worst case and the best case.
- C) Worst case complexity of an algorithm can be determined by choosing a set of input conditions that force the algorithm to make the least possible progress towards its final goal at each step.
- D) Algorithms having exponential complexity can be practically solved only for very small values of n.

1.2 The correct ADT of pop operation of a stack is

- A) Function: Adds *newItem* to the top of stack
Precondition: Stack has been initialized
Postcondition: If (stack is full), exception fullStack is thrown,
else *newItem* is at the top of the stack
- B) Function: Adds *newItem* to the top of stack
Precondition: Stack has been initialized
Postcondition: If (stack is empty), exception emptyStack is thrown,
else top element is removed from the stack
- C) Function: Removes top element from the stack
Precondition: Stack has been initialized
Postcondition: If (stack is empty), exception emptyStack is thrown,
else top element is removed from the stack
- D) Function: Removes top element from the stack
Precondition: Stack has been initialized
Postcondition: If (stack is full), exception FullStack is thrown,
else *newItem* is at the top of the stack

1.3 The result of $4 \cdot 5 + 6 \cdot 1 - 2$ is

- A) 18
- B) -18
- C) 17
- D) -17

- 1.4 Which of the following statement is valid to define a dynamic array of 10 elements?
- A) `int *p = new int[10];`
 - B) `int p = new int[10];`
 - C) `int a[10];`
 - D) `int a[] = new int[10];`

- 1.5 Identify the correct statement from the following:
- A) $O(n^3) < O(n^2 \log_2 n)$
 - B) $O(2^n) < O(n^2)$
 - C) $O(n) > O(\log_2 n)$
 - D) $O(n^2) > O(n^3)$

- 1.6 Consider the following matrix

$$A = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

If array A is stored in row major order (without storing 0's), then element a_{ij} will be stored at location with the index (the starting value of the index is 1)

- A) $\frac{j(j+1)}{2} + i$
- B) $\frac{j(j-1)}{2} + i$
- C) $\frac{i(i+1)}{2} + j$
- D) $\frac{i(i-1)}{2} + j$

- 1.7 When the data is already sorted, then _____ has $O(n)$ complexity.
- A) mergesort
 - B) quicksort
 - C) insertion sort
 - D) none of the sorting algorithms have $O(n)$ complexity for any case

- 1.8 Minimal spanning tree may be a forest at intermediate stages in _____ algorithm.
- A) Prim's
 - B) Kruskal's
 - C) Both Prim's and Kruskal's
 - D) Neither Prim's nor Kruskal's

- 1.9 If linear search is unsuccessful then the number of comparisons done is
- A) $n / 2$
 - B) n
 - C) $n + 1$
 - D) $n - 1$

- 1.10 The maximum number of nodes possible at n^{th} level of a binary tree is _____ (assuming root to be at level 0).
- A) 2^n
 - B) 2^{n+1}
 - C) 2^{n-1}
 - D) n^2

2. Each statement below is either TRUE or FALSE. Choose the most appropriate one and ENTER in the “tear-off” sheet attached to the question paper, following instructions therein. (1x10)

- 2.1 Binary search is not feasible if data is stored in a linked list.
- 2.2 A tree with n nodes has n + 1 edges.
- 2.3 Data can be removed from either sides of a D-queue.
- 2.4 Inorder traversal of any binary tree results in ordered data.
- 2.5 The time complexity of some algorithms can be decreased if more space is used.
- 2.6 If the data is sorted then even without linearly searching the entire list it can be inferred that the target being searched is not found.
- 2.7 Graphs are special types of trees.
- 2.8 Array elements are stored contiguously in memory.
- 2.9 The prefix and postfix expressions are mirror images of each other.
- 2.10 Trees are non-linear data structures.

3. Match words and phrases in column X with the closest related meaning/ word(s)/phrase(s) in column Y. Enter your selection in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)

X		Y	
3.1	Binary search trees	A.	Breadth first search
3.2	Expressions without brackets	B.	Rear
3.3	Bubble sort	C.	Root
3.4	Classes	D.	goto-less programming
3.5	Special node	E.	Programming with structures
3.6	Complexity measure	F.	Big-O
3.7	Binary Search	G.	Postfix
3.8	Queue	H.	Divide and conquer strategy
3.9	Structured Programming	I.	Infix
3.10	Graphs	J.	Consecutive elements comparison
		K.	Right subtree data greater than or equal to the one in the node
		L.	Polymorphism
		M.	Encapsulation

4. Each statement below has a blank space to fit one of the word(s) or phrase(s) in the list below. Enter your choice in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)

A.	Operand stack	B.	Top-down design	C.	Radix
D.	Operator stack	E.	FIFO	F.	LIFO
G.	Sparse	H.	O(1)	I.	Linked Lists
J.	Main memory	K.	Trees	L.	Threaded binary
M.	Complete binary				

- 4.1 A matrix having zero elements except the diagonal elements is known as a _____ matrix.
- 4.2 In _____ sort algorithm, the elements of the list are not compared with each other.
- 4.3 _____ trees have some nodes pointing to the ancestors in the tree.
- 4.4 Complexity of inserting an element in a queue, currently having n elements is _____.
- 4.5 Circular queues may be implemented using _____ also.
- 4.6 _____ is used to convert an infix expression to its corresponding postfix form.
- 4.7 Queues follow _____ policy.
- 4.8 In a _____ tree, a node if it has right child will have the left child also.
- 4.9 The logical or mathematical model of organization of data in the _____ is called a data structure.
- 4.10 The process of repeatedly breaking a problem into subproblems till they can be solved is _____.

PART TWO
(Answer any **FOUR** questions)

5.

- a) Create a class *Matrix* having the following:

Data members

row, column and an array to store the elements of the matrix

Function members

* A function to multiply two matrices

* A function to display the contents of the matrix in rectangular form

Assume appropriate methods exist in the class to read the contents of the matrices.

- b) Show that the function $L(n)$ defined by

$$L(n) = \begin{cases} L\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 & \text{if } n > 1 \\ 0 & \text{if } n = 0 \end{cases}$$

has the complexity $O(\log_2 n)$.

(9+6)

6.

- a) Convert the following infix expression to postfix form using stack: (Describe the stack at every stage)

$$(A + B * C) / (D - E) + F$$

- b) What is polymorphism? Briefly explain the different types of polymorphism supported in C++.
c) Write a C++ function to reverse a singly linked list by traversing it only once.

(5+5+5)

7.

- a) An AVL tree is implemented such that every node has a pointer to its parent. Design a procedure to insert and another to delete a node from the AVL tree. You must take care of the height balance and empty tree.

- b) Construct the binary tree for the following sequence of traversals of nodes:

Inorder: E A C K F H D B G

Preorder: F A E K C D H G B

- c) What is the difference between a binary tree and a binary search tree? Give an example of tree that is binary tree but not a binary search tree.

(8+4+3)

8.

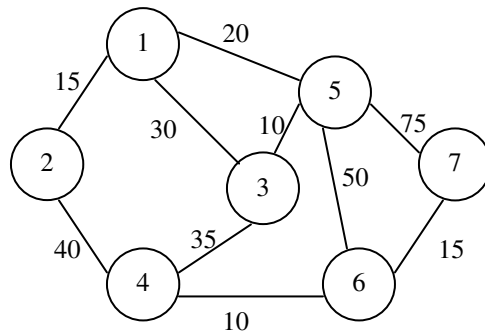
- a) The following keys are to be inserted in the order into a heap: 25, 57, 48, 37, 9, 97, 86, 33. Show how the tree appears after each insertion.

- b) Write a procedure to traverse a Binary search tree in such a manner that the values at nodes are printed in reverse (descending order of value).

(8+7)

9.

- a) Differentiate between a stack and a queue. Define the class *Queue* to implement a circular queue using arrays.
- b) Write the adjacency matrix of the following graph:



- c) Using Kruskal's algorithm, find the minimal spanning tree of the graph given in Q9 b). Show the intermediate steps also.

(7+3+5)