

23. a. Write an algorithm to convert an infix expression to a postfix expression. Trace the algorithm to convert the infix expression $a+b * c/d+e/f$ to a postfix expression. 8 4 3 3

(OR)

- b. Give algorithms for push and pop operations on stack using a linked list with suitable pictorial representations. 8 3 3 2

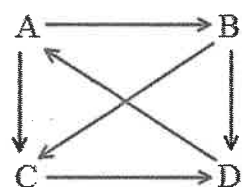
24. a. Write appropriate algorithm for inserting and deleting a node in a binary search tree. 8 4 3 3

(OR)

- b.i. Create a B+tree of order – 5 for the following data arriving in sequence 90, 27, 7, 9, 18, 21, 3, 4, 16, 11, 21, 72. 4 4 4 2

- ii. Compare BST with AVL tree. 4 3 4 2

25. a. Consider the graph given below show its adjacency list and adjacency matrix in the memory. 8 5 5 2



(OR)

- b. Illustrate Kruskal's algorithm to find the minimum spanning tree of a graph. 8 4 5 2

PART – C (1 × 15 = 15 Marks)

Answer ANY ONE Questions

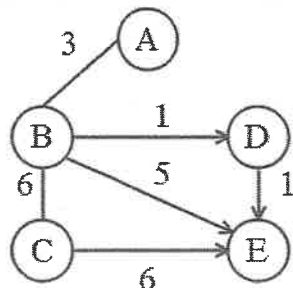
Marks BL CO PO

26. Consider a hash table with 9 slots. The hash function is $h(k) = k \bmod 9$. The following keys are inserted in order 5, 28, 19, 15, 20, 33, 12, 17, 10.

Draw the contents of the hash table when the collisions are solved by

- Chaining
- Linear probing
- Double hashing. The second hash function $h_2(x) = 7 - (x\%7)$.

27. Apply an appropriate algorithm to find the shortest path from 'A' to every other node of A. For the given graph.



* * * * *

Reg. No.

B.Tech. DEGREE EXAMINATION, MAY 2023

Third Semester

21CSC201J – DATA STRUCTURES AND ALGORITHMS

(For the candidates admitted from the academic year 2022-2023 onwards)

Note:

- Part - A** should be answered in OMR sheet within first 40 minutes and OMR sheet should be handed over to hall invigilator at the end of 40th minute.
- Part - B** and **Part - C** should be answered in answer booklet.

Time: 3 Hours

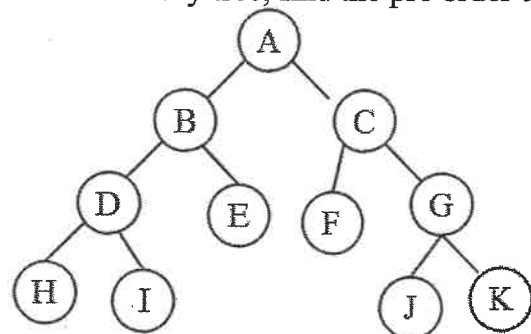
Max. Marks: 75

PART – A (20 × 1 = 20Marks)

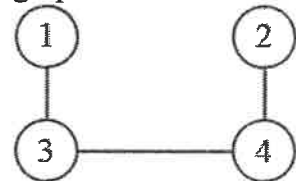
Answer ALL Questions

- | | Marks | BL | CO | PO |
|--|-------|----|----|----|
| 1. How do you instantiate an array in C?
(A) <code>int Arr[] = New int (3);</code> (B) <code>int Arr [3];</code>
(C) <code>int Arr[] = new int[3];</code> (D) <code>int Arr() = New int [3];</code> | 1 | 1 | 1 | 1 |
| 2. Elements in an array are accessed
(A) Randomly (B) Sequentially
(C) Exponentially (D) Logarithmically | 1 | 1 | 1 | 1 |
| 3. Which of the following data structure is non linear type?
(A) Strings (B) Lists
(C) Stacks (D) Hierarchical | 1 | 1 | 1 | 3 |
| 4. What will be the output of the following code?
<pre> struct student { CHAR *C; }; void main() { struct student *S ; S->C = "HELLO"; printf("%S", S->C); } </pre> (A) Hello (B) Segmentation fault
(C) Runtime error (D) Nothing | 1 | 2 | 1 | 1 |
| 5. Linked list data structure offers considerable saving in _____
(A) Computational time (B) Space utilization
(C) Space utilization and (D) Speed utilization computational time | 1 | 1 | 1 | 1 |
| 6. Which of the following is false about a doubly linked list?
(A) We can navigate in both the directions (B) It requires more space than a singly linked list
(C) The insertion and deletion of a node take a bit longer time (D) Implementing a doubly linked list is easier than singly linked list | 1 | 1 | 2 | 1 |

7. Which of the following is false about a circular linked list? 1 1 2 1
 (A) Every node has a successor (B) Time complexity of inserting a new node at the head of the list is $O(1)$
 (C) The complexity for deleting the last node is $O(n)$. (D) We can traverse the whole circular linked list, starting from any point
8. Which of these is not an application of a linked list? 1 2 2 2
 (A) To implement file system (B) For separate chaining in hash-tables
 (C) To implement non-binary trees (D) Random access of elements
9. The data structure required to check whether an expression contains balanced parenthesis is 1 1 3 3
 (A) Stack (B) Queue
 (C) Array (D) Tree
10. The prefix form of $A-B / (C*D \wedge E)$ is? 1 2 3 3
 (A) $-/* \wedge ACBDE$ (B) $-ABCD * \wedge DE$
 (C) $-A/B * C \wedge DE$ (D) $-A/BC * \wedge DE$
11. A queue follows _____ 1 1 3 3
 (A) FIFO (First in First Out) principle (B) LIFO (Last in First Out) principle
 (C) Ordered array (D) Linear tree
12. A normal queue, if implemented using an array of size max-size gets full when? 1 1 3 2
 (A) $Rear = Max - size - 1$ (B) $Front = (rear + 1) \bmod \text{max size}$
 (C) $Front = Rear + 1$ (D) $Rear = Front$
13. A linear list of elements in which deletion can be done from one end (front) and insertion can take place only at the other end (rear) is known as _____. 1 1 3 1
 (A) Queue (B) Stack
 (C) Tree (D) Linked list
14. What is a full binary tree? 1 1 4 1
 (A) Each node has exactly zero or two children (B) Each node has exactly two children
 (C) All the leaves are at the same level (D) Each node has exactly one or two children
15. For the binary tree, find the pre order traversal sequence. 1 1 4 3



- (A) H I D E G F J K G C A (B) H D I B E A F C J G K
 (C) A B D H I E C F G J K (D) A B C D E F G H I J K

16. What is direct addressing? 1 1 4 1
 (A) Distinct array position for every possible (B) Fewer array positions than keys
 (C) Fewer keys than array positions (D) Same array positions for all keys
17. What data organization method is used in Hash tables? 1 1 4 3
 (A) Stack (B) Array
 (C) Linked list (D) Queue
18. What would be the number of zeros in the adjacency matrix of the given graph? 1 1 5 3

 (A) 10 (B) 6
 (C) 16 (D) 0
19. The traveling salesman problem can be solved using _____. 1 1 5 1
 (A) A spanning tree (B) A minimum spanning tree
 (C) Bellman – Ford algorithm (D) DFS traversal
20. DIJKSTRA's algorithm is used to solve _____ problem. 1 1 5 1
 (A) All pair shortest path (B) Single source shortest path
 (C) Network flow (D) Sorting

PART – B (5 × 8 = 40 Marks)

Answer ALL Questions

21. a. Define a function to multiply two $m \times n$ matrices and return the resultant matrix. 8 4 1 9
- (OR)
- b. How mathematical notations are used to measure the complexity of the algorithm? Explain different notations used. 8 3 1 3
22. a. Write an algorithm to demonstrate a polynomial equation using linked list for subtraction of two polynomials equations. 8 3 2 2

(OR)

- b. Consider an array $A[1:n]$ Given a position, write an algorithm to insert an element in the array. If the position is empty, the element is inserted easily. If the position is already occupied the element should be inserted with the minimum number of shifts (Note: The elements can shift to the left or to the right to make the minimum number of moves). 8 4 2 3