Reg. No. : E N G G T R E E . C O M

Question Paper Code : 40485

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2024.

Second/Third Semester

Computer Science and Design

CD 3291 — DATA STRUCTURES AND ALGORITHMS

(Common to : Computer Science and Engineering (Artificial Intelligence and Machine Learning)/ Computer Science and Engineering (Cyber Security)/ Computer and Communication Engineering/Information Technology)

(Regulations 2021)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Differentiate between shallow copying and deep copying.

2. Find the time complexity of the following recurrence relation :

   $T(n)=1+T(n-1)$ if $n>1$

   $T(1)=0$.

3. List any two applications of a queue.

4. What is the disadvantage of a circular linked list?

5. What is the computational complexity of bubble sort in the best and worst case?

6. What is the purpose of hashing?

7. Construct a max heap from the following data: 12, 4, 32, 45, 6.

8. A binary tree T has 9 nodes. From the inorder and preorder traversals of T given below, construct the binary tree.

   Inorder traversal : E A C K F H D B G

   Preorder traversal : F A E K C D H G B

9. Find the topological ordering of the following graph (Fig. 9)

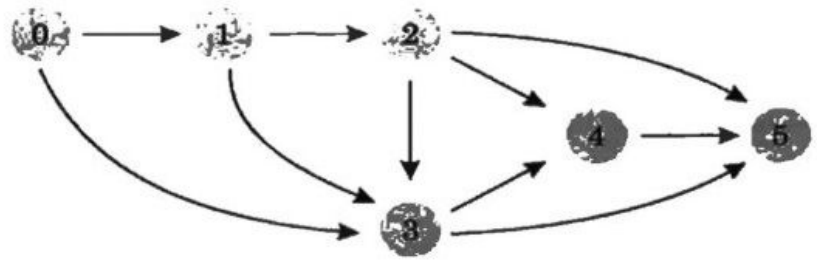

Fig. 9

10. Distinguish tractable and intractable algorithms.

PART B — (5 × 13 = 65 marks)

11. (a) (i) What is operator overloading? Write a program to overload '+' operator to concatenate two strings. (6)

(ii) ALGORITHM $Q(n)$

//Input: A positive integer n

if $n = 1$ return 1

else return $Q(n-1) + 2 * n - 1$

Identify the basic operation in the algorithm. Set up the recurrence relation for the number of times the algorithm's basic operation is executed and solve it using back substitution. (7)

Or

(b) Write an algorithm to perform binary search and analyse its time complexity.

12. (a) Write procedures for performing the following operations on a singly linked list.

(i) Delete a node at any arbitrary location. (6)

(ii) Insert a node at the end. (7)

Or

(b) Write an algorithm to convert an infix expression to a postfix expression. Trace the algorithm for the infix expression A+(B*C–(D/E^F)*G)*H. Evaluate the obtained postfix expression with A=1, B= –1, C=3, D=8, E=2, F=2, G= –5, H=4. Use stack for both operations.

13. (a) Write an algorithm to perform merge sort and analyse its worst-case time complexity. Apply it to sort 12, 34 ,5, 67, 24, 10. (13)

Or

(b) Define Hashing. Insert the following data into a hash table using hash function h(X) = X mod 9. {17, 9, 34, 56, 11, 71, 86, 55, 22, 10, 4, 39, 49, 52, 82, 13, 40, 31, 35, 28, 44} for each of the following scenarios.

- Collisions are handled by linear probing (slots = 3) (7)
- Collisions are handled by separate chaining. (6)

14. (a) Insert the following keys into an initially empty binary search tree (insert the items in the given order) : 30, 40, 23, 58, 48, 26, 11, 13. From the resultant tree, delete the following keys in sequence : 40, 58, 30, 23. Show the tree after each operation. (13)

Or

(b) Explain the significance of AVL trees over binary search trees and give the generic representations of LL, and RL operations. Insert the elements 8, 9, 12 and 10 in the AVL tree given in Fig. 14(b). Delete the following elements 8, 2, and 7 from the tree in sequence. Show the tree after each operation and mention the type of rotation in each case. (13)

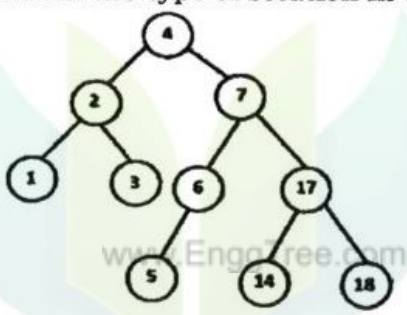

Fig. 14(b)

15. (a) For the following graph Fig. 15(a)

(i) Report the order of the vertices encountered on a breadth-first search starting from vertex A. (6.5)

(ii) Report the order of the vertices encountered on a depth-first search starting from vertex A.

Break all ties by picking the vertices in alphabetical order. (6.5)

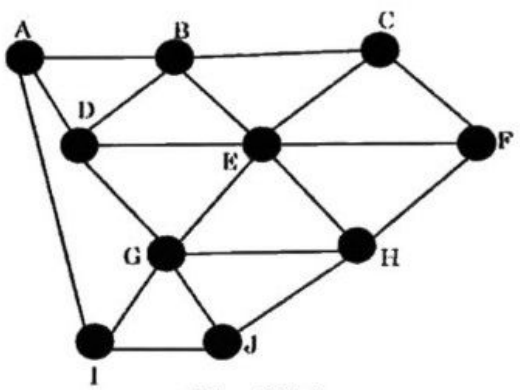

Fig. 15(a)

Or

(b) Write Prim's algorithm to find the minimum spanning tree. Apply the algorithm to find the minimum spanning tree of the graph given in Fig. 15(b).

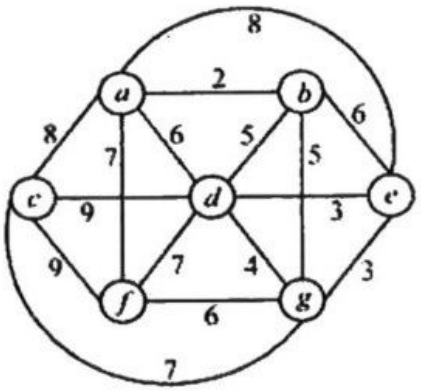

Fig. 15(b)

PART C — (1 × 15 = 15 marks)

16. (a) List the properties of m-way search tree. Explain the procedure for insertion and deletion of a key in a m-way search tree. Starting with an empty 4-way tree, perform the following operations in sequence : Insert 40, 80, 20, 60, 70, 45, 88, 120, 37, 21, 22 and Delete 45, 60, 22, 40, 88.

Or

(b) Write the pseudocode for Dijkstra's algorithm to solve the single-source shortest-path problem on a weighted, directed graph assuming all edge weights are nonnegative. Analyze the algorithm to find the time complexity. Trace the algorithm on the following graph in Fig. 16(b) by showing each execution step of the algorithm. Assume the source vertex as 'A'

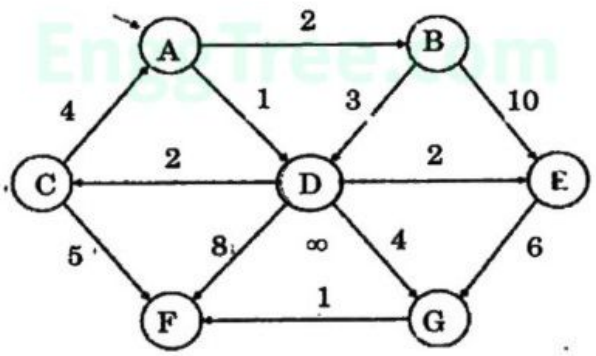

Fig. 16(b)