Total No. of Questions—**12**] [Total No. of Printed Pages—**4+1**

[3662]-267

S.E. (Comp. Engg.) (Second Semester) EXAMINATION, 2009

DATA STRUCTURES

(2003 COURSE)

Time : Three Hours **Maximum Marks : 100**

N.B. :— (*i*) Answer *three* questions from Section I and *three* questions from Section II.

(*ii*) Answers to the two Sections should be written in separate answer-books.

(*iii*) Neat diagrams must be drawn wherever necessary.

(*iv*) Figures to the right indicate full marks.

(*v*) Assume suitable data if necessary.

SECTION I

1. (*a*) Write pseudo 'C' code to perform insertion of a node in singly linked list maintained on non-decreasing order. Your algorithm should consider all possibilities. [6]

(*b*) Implement circular queue using circular singly linked list. [6]

(*c*) Write pseudo 'C' code to perform deletion of a node of given value from circular doubly linked list. [4]

Or

2. (*a*) Define suitable data structure to represent multi-variable polynomial using generalized linked list and represent the following

P.T.O.

polynomial using the node structure defined by you : [6]

$$P(x, y, z) : x^5 y^3 z^3 + 3x^3 y^3 z^3 + 9x^2y^2z^2 + 24x^2yz + 8xyz.$$

(*b*) Write pseudo 'C' code to update number of nodes to the right of each node in a given singly linked list. [6]

(*c*) Write pseudo 'C' code to revert the given singly linked list only by manipulation of pointers. [4]

3. (*a*) Write non-recursive pseudo 'C' code to traverse a given binary tree in pre-order. What is the time complexity of your algorithm ? [6]

(*b*) Generate binary tree for the following pre-order and in-order traversals : [6]

Pre-order : 1 2 4 7 3 5 6 8 9

In-order : 7 4 2 1 5 3 8 6 9

(*c*) Write pseudo 'C' code to delete a given node from a given binary search tree. [4]

Or

4. (*a*) Write pseudo 'C' code to insert and to delete a given node (by value) from a given in-order threaded binary tree. [8]

(*b*) Explain in detail with examples any *two* applications of binary tree. [8]

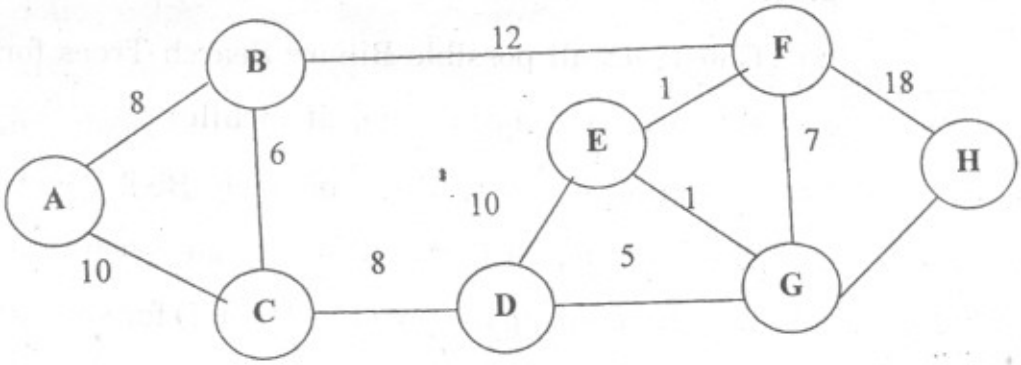
5. (*a*) Write pseudo 'C' code to print level wise nodes of a spanning tree generated by traversing a given graph 'G' containing '*n*' vertices. What is the time complexity of your algorithm ? [6]

(*b*) Write pseudo 'C' code to get minimum spanning tree from a given graph 'G' using Kruskal's algorithm. Compare it with Prim's algorithm and comment on their usage. [8]

(*c*) State any *four* applications of graphs. [4]

Or

6. (*a*) Obtain minimum spanning tree for the following graph using Prim's algorithm. Show stepwise results of your solution in tabular form : [6]



(*b*) Write pseudo 'C' code to get minimum spanning tree from a given graph 'G' using Prim's algorithm. What is its time complexity. [6]

(*c*) For the graph given in Q-6(*a*), obtain shortest paths from vertex A to all other vertices using Dijkshtra's algorithm. Show step-wise results of your solution in tabular form. [6]

SECTION II

7. (*a*) What is AVL tree ? Why is it necessary to balance the height of the tree ? Construct AVL tree for the following data. Identify each rotation while constructing the AVL tree : [10]

21, 97, 85, 78, 74, 63, 45, 16, 52, 19.

(*b*) What do you understand by Collision in hashing ? Represent the following keys in memory using linear probing with and without replacement. Use modulo (10) as your hashing function : [8]

(24, 13, 16, 15, 19, 20, 22, 14, 17, 26, 84, 96)

Or

8. (*a*) How many binary search trees (BSTs) can be constructed for given '*n*' identifiers ?

(*i*) Construct all possible Binary Search Trees for the identifier set (a_1, a_2, a_3) = (do, if, while)

(*ii*) Compute the total cost of each BST constructed by you assuming equal probabilities for successful and unsuccessful search i.e. $(p_1, p_2, p_3) = (1/7, 1/7, 1/7)$ for successful search and $(q_0, q_1, q_2, q_3) = (1/7, 1/7, 1/7, 1/7)$ for unsuccessful search. Which BST is an optimal binary search tree ?

(*iii*) Compute the total cost of each BST constructed by you assuming probabilities for successful and unsuccessful search as :

$(p_1, p_2 p_3) = (0.5, 0.1, 0.05)$ for successful search and $(q_0, q_1, q_2, q_3) = (0.15, 01, 0.05, 0.05)$ for successful search. Which BST is an optimal binary search tree ? [10]

(*b*) What is Hashing ? Represent the following keys in memory using chaining with and without replacement. Use Modulo (10) as your hashing function : [8]

(52, 11, 84, 63, 77, 18, 40, 12, 25, 29, 32, 65)

9. (*a*) Write a function in 'C' to create a Max Heap for the given array containing '*n*' identifiers. Obtain time and space complexity of your algorithm. [8]

(*b*) Write short notes on the following : [8]

(1) B-tree

(2) Trie Indexing.

Or

10. (*a*) Define Max Heap. Write pseudo 'C' code to implement priority queue using max heap. Assume that max heap is already constructed in memory. Obtain time complexity for addition and deletion of an element from priority queue. [8]

(*b*) Write short notes on the following :

(1) Red-black tree

(2) k-d tree. [8]

11. (*a*) What are external storage devices ? Explain each one in brief. [6]

(*b*) What type of file organization is used to handle multiple keys ? Explain in detail. [6]

(*c*) Explain how records are accessed randomly in case of random file. [4]

Or

12. (*a*) What is blocking factor in case of tape files ? Can tap files have blocking factor as 1 ? Is it relevant in case of disk files ? Explain. [6]

(*b*) Write 'C' implementation of the primitive operations for index sequential file organization. [6]

(*c*) Write a short note on Cellular partitioning. [4]