BTS-IV-04.19-0669 Reg. No. | | | | | | | | | B

B.Tech. Degree IV Semester Examination April 2019

CS/IT 15-1405 DATA STRUCTURES AND ALGORITHMS

(2015 Scheme)

Time: 3 Hours Maximum Marks: 60

PART A

(Answer *ALL* questions)

(10 × 2 = 20)

I. (a) What are sparse matrices? How can sparse matrices be efficiently represented?

(b) State the advantage of binary search over linear search.

(c) State two applications of queue as a data structure.

(d) What is the output of the following function where 'head' is pointing to the first node of the following linked list?

Linked list: 1→2→3→4→5→6

```
void fun(Node head)
{
    if(head == NULL)
        return;
    fun(head.next);
    System.out.println(head.data);
}
```

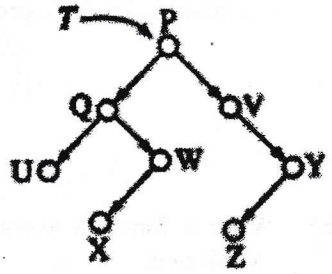
(e) Evaluate the following postfix expression using stack, assuming each operand is a single digit positive integer and each operator is binary.

4 2 3 + - 3 8 2 / + *

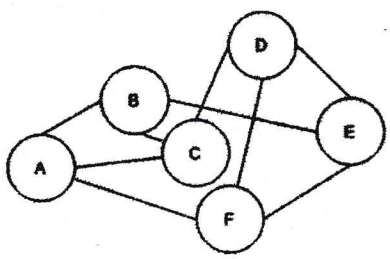
(f) How is an AVL tree better than a binary search tree?

(g) Write short note on Threaded Binary Tree (TBT).

(h) Consider the following binary search tree T as given below. Which node contains the fourth smallest element in T?



(i) Is the given graph regular? Justify your answer.



(j) Differentiate between B Trees and B+ Trees.

(P.T.O.)

PART B

(4 × 10 = 40)

II. (a) Insert the characters of the string **K R P C S N Y T J M** into a hash table of size 10 . Use hash function **h(X) = (ord(X) – ord('A') + 1) mod 10** , where **ord(X)** refers to the relative position of **X** in the alphabetic array (Eg: ord(A)=1,ord(B)=2..ord(Z)=26). (5)

Use linear probing to resolve collision.

i. Which insertions cause collision?

ii. Display the final hash table.

(b) Write the algorithm for quick sort. Illustrate with an example (5)

OR

III. Consider the following elements in an array: 23,33,12,55,38,66,59. Perform Bubble sort, Insertion Sort and Selection Sort on the input data. Show the status of array after each iteration. (3 + 3 + 3 + 1 = 10)

Why are the above mentioned sorting algorithms named so?

IV. (a) Implement a stack using an array. You need to write the pseudo-code for all the primitive operations of the stack. (5)

(b) Given two singly linked lists L1 and L2. Write pseudo-code to merge both linked lists to a single linked list. (5)

OR

V. Explain with an algorithm, how infix expression can be converted into postfix form. Illustrate it with the expression: A+B-C*D+(E^F)*G/H/I*J+K. (10)

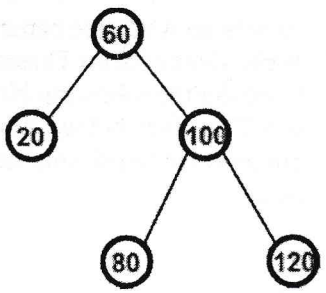
VI. (a) Draw a binary tree for the expression **4+3*7-5/(3+4)+6.** (4)

(b) Traverse the above generated tree using inorder, preorder and postorder traversals (3)

(c) Also write a recursive function for each of the three traversal methods (3)

OR

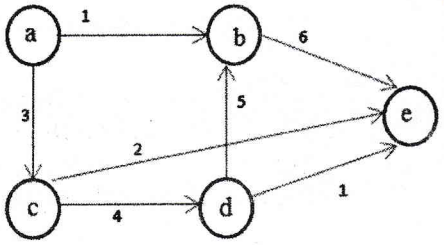
VII. (a) Show intermediate steps to insert the node 70 into the given AVL tree maintaining its AVL property? (4)



(b) Write a function (Java code/pseudo-code) to insert a node into a binary search tree. (6)

(Contd.....3)

VIII. (a) Consider the following graph and find the shortest path from node 'a' to node 'e' using Dijkstra's algorithm. Show all the steps involved. (4)

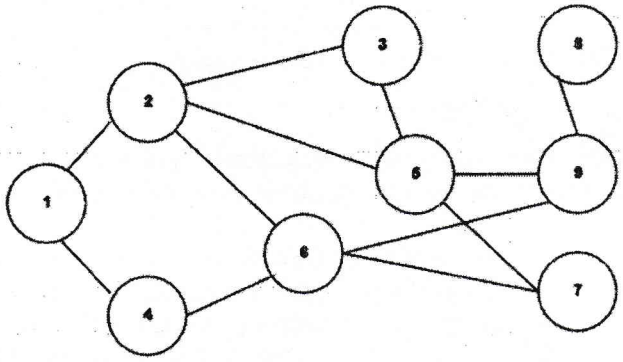


(b) What is a spanning tree? Consider an undirected graph of vertices{A,B,C,D,E,F}. Adjacency matrix of the graph is given below. Find the minimum spanning tree using Prim's algorithm. Show every step of action. (6)

$$\begin{pmatrix} 0 & 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 6 & 2 & 0 \\ 2 & 3 & 0 & 5 & 0 & 3 \\ 0 & 6 & 5 & 0 & 4 & 1 \\ 0 & 2 & 0 & 4 & 0 & 0 \\ 1 & 0 & 3 & 1 & 0 & 0 \end{pmatrix}$$

OR

IX. Explain the rules for BFS and DFS. What are the data structures used for each of the traversal method? Why are they used? (10)
Demonstrate with the given graph, how each traversal is being carried out, where the start node is 1.



***