Total No. of Questions : 4]

SEAT No. :

PF212

[Total No. of Pages : 2

APR-26/SE/Insem-261

S.E. (Computer Engg./AI & DS/Computer Science)

DATA STRUCTURES AND ALGORITHMS

(2019 Pattern) (Semester - IV) (210252)(Insem)

Time : 1 Hour] *[Max. Marks : 30*

Instructions to the candidates:

1) ***Solve Q.1 or Q.2 and Q.3 or Q.4.***
2) ***Neat diagrams must be drawn wherever necessary.***
3) ***Figures to the right indicate full marks.***
4) ***Assume Suitable data if necessary.***
5) ***Use of non-programmable calculator is allowed.***

Q1) a) For the given set of values. **[6]**

11, 33, 20, 88, 79, 98, 44, 68, 66, 22

Create a hash table with size 10 and resolve collision using chaining with replacement. Use the modulus Hash function. (key % size.)

b) Insert the keys 9, 19, 29, 39, 49, 59, 71 into the Hash Table of size 10. Resolve all collisions using Quadratic probing where hash-function is *h*(k). h(k) = k mod 10 **[5]**

c) Write difference between separate chaining and open addressing. **[4]**

OR

Q2) a) Solve the following example using extendible hashing. **[6]**

Elements: 28,4,19,1,22,16,12,0,5,7 & Bucket Size: 3

b) Construct hash table of size 10 using linear probing without replacement strategy for collision resolution. The hash function is h(x)=x%10. Consider slot per bucket is 1. 31, 3, 4, 21, 61, 6, 71, 8, 9, 25. **[5]**

c) What is Skip List? What are advantages and disadvantages of skiplist?**[4]**

P.T.O.

Q3) a) Write an algorithm for recursive and non-recursive postorder traversal of binary tree and give suitable example. **[6]**

b) Construct Huffman's Tree and the prefix free code for all characters: **[5]**

Value	A	B	C	D	E	F
Frequency	5	25	7	15	4	12

c) From the given traversals construct the binary tree. **[4]**

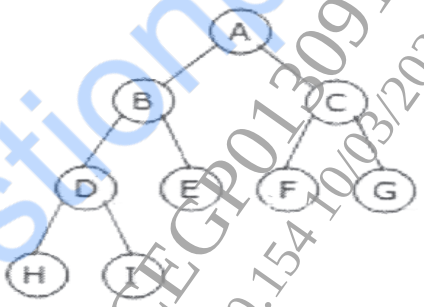
Pre-order: G, B, Q, A, C, K, F, P, D, E, R, H

In-order: Q, B, K, C, F, A, G, P, E, D, H, R

OR

Q4) a) Write an algorithm to delete node from BST with example. **[6]**

b) Explain the storage representation of a binary tree with the following example. **[5]**



c) Explain Binary Search Tree. Construct Binary Search Tree(BST) for the following : J, R, D, G, T, E, M, H, P, A, F, Q. **[4]**