Roll No. .....

Total Pages : 3

311302

December 2022

BCA- III Semester

Data Structures (BCA-17-202)

Time : 3 Hours] [Max. Marks : 75

Instructions :

1. *It is compulsory to answer all the questions (1.5 marks each) of Part-A in short.*
2. *Answer any four questions from Part-B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

PART–A

1. (a) Define data structures. (1.5)

(b) What is algorithm? (1.5)

(c) Explain what is time-space trade-off ? (1.5)

(d) Differentiate between linear and non-linear data structures. (1.5)

(e) What is circular linked list? (1.5)

(f) What is operand stack and what is it used for? (1.5)

(g) Define recursion. (1.5)

(h) Is the following statement True or False? (1.5)

"A unique binary tree can be created if its post order and pre order traversal are given."

(i) Explain post order tree traversal. (1.5)

(j) Explain depth first search. (1.5)

PART–B

2. (a) Explain three main asymptotic notations for time complexity with diagram. (10)

   (b) Explain the naive algorithm for pattern matching in strings with example.. (5)

3. (a) Compare singly linked list with doubly linked list with diagram. (5)

   (b) Suppose there is a 2-D array of size 10×20: int a [10][20] stored at the base address of 2058. The range of the two-dimensions are [5 ..... 14] [0 ..... 19]. If the system follows row major ordering, then calculate the address of a [9] [16]. (10)

4. Explain double ended queues with a diagram. Write the algorithm for implementing dequeue operation using stacks. (15)

5. (a) Write down the algorithm for enqueue operation in a queue. (5)

   (b) Convert following infix notation to postfix notation using operator stack-Infix notation

   $= a \wedge e - g / b \wedge d * c.$ (10)

6. (a) Construct a binary search tree from given preorder traversal.

   Preorder : 25, 20, 15, 10, 13, 18, 50, 30, 55, 40. (10)

   (b) Define binary search trees. What is the difference between binary trees and binary search trees. (5)

7. Find out single source shortest paths for the source 'S' using Dijkstra's algorithm. (15)