F 9358 (Pages : 2) Reg. No.....

Name.....

B.TECH. DEGREE EXAMINATION, NOVEMBER 2011

Fourth Semester

Branch : Computer Science and Engineering

DATA STRUCTURES AND PROGRAMMING METHODOLOGIES (R)

(2002 Admissions onwards —Supplementary)

Time : Three Hours Maximum : 100 Marks

Part A

Answer **all** *questions.*
Each question carries 4 *marks.*

1. Give a recursive algorithm to reverse a character string.
2. Define "time complexity of an algorithm". Why are computer scientists interested in it ?
3. What is a priority queue ? How can it be implemented efficiently ?
4. A circular queue is implemented using an array of size n. Give a formula in terms of *front*, *rear* and n for the number of elements in the queue. How does the queue become circular ?
5. Compare and contrast arrays with linked lists.
6. Give a function to reverse a singly linked list.
7. Give a recursive function to search a graph in Depth First order.
8. Prove that a binary tree of height h ($h >= 0$) has a least h and at most $2^h - 1$ elements in it.
9. Briefly explain the process of external sorting using merge sort.
10. What is heap ? Explain its properties.

(10 × 4 = 40 marks)

Part B

Answer **all** *questions.*
Each question carries 12 *marks.*

11. Write an algorithm to find the n-th Fibonacci number, and find out its time complexity.

Or

12. Give a recursive version of Binary Search algorithm and derive its worst case time complexity.
13. A list is being maintained as a circular queue, which is implemented using an array of size n.
    (a) Write an algorithm to delete the k-th element in the list.
    (b) Write an algorithm to insert an element immediately after the k-th element in the list.

Or

14. What is a Dqueue ? How do you implement it using array ? Write codes to enqeue and dequeue elements.

Turn over

15. Write code/pseudo-code to implement a queue using circular linked list and explain its working

Or

16. Write a code/pseudo-code for adding two polynomials represented as singly linked lists. Explain its working.

17. Give and explain the working of non-recursive Depth First Search and Breadth First Search algorithms.

Or

18. Write a function/pseudo-code for :

    (a) Counting the number of leaf nodes.

    (b) Swapping the left and right child of each node of a Binary Tree, given the pointer to the root.

19. Give the Heap Sort algorithm. Trace the working with a sample set of 10 numbers and draw all the intermediate heaps formed.

Or

20. Write a program/pseudo-code for Quick Sort. Analyze its performance and time complexity when an input, which is already sorted in descending order, is given.

(5 × 12 = 60 marks)