THIRD SEMESTER EXAMINATION-2012

DATA STRUCTURES & ALGORITHM

[CS 301]

Full Marks: 60 Time: 3 Hours

Answer any SIX questions including Question No.1 which is compulsory.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

1. a) Briefly discuss how a sparse matrix can be used effectively? [2 × 10

   b) Construct a two-way inorder threaded binary search tree with the following set of elements.

      12, 3, 45, 10, 8, 11, 67, 80, 59, 26, 35

   c) Tree data structure is an ADT. (True/False) Justify.

   d) Construct a binary tree with the following sequence of traversal given. The English alphabets represent the node information.

      Preorder: ABCDFHJMKEGILN

      Inorder : ADJMHKFCINLGEB

   e) Bubble sort algorithm is inefficient because it continues execution even after an array is sorted by performing unnecessary comparisons. Therefore, the number of comparisons in the best and worst cases is the same. Modify the algorithm in such a fashion that it will not make the next pass when the array is already sorted.

f) Write a pseudo code to find the maximum element from a stack in O(1) time at the time of creation of the stack.

g) Under which situation the best case and worst case arises incase of insertion sort algorithm? Write down the respective time complexity.

h) How many queues ADT are required to represent a stack ADT? Justify your answer with suitable example.

i) Represent all data structures you have studied in a tabular format and give its advantages, disadvantage and application.

j) What are the necessary conditions required to perform binary search? Write a recursive binary search algorithm.

2. a) Differentiate between a binary tree and a B-tree. Discuss the technique used to minimize the access time in both the cases. Construct a B-tree of order 3 with the following elements(show all steps): [4

21,23,36,45,19,8,15,41,52,75,66,11,90,84,1

b) What is polish notation? Write an algorithm to convert an infix expression into a postfix expression using stack ADT. Explain the algorithm with the given infix expression: [4

a+b-c*(d*e)-f+g+h

3. a) What is a queue data structure? Write an algorithm to implement insertion and deletion operation incase of an input-restricted deque and output-restricted deque. [4

b) Suppose V_1, V_2, V_3, V_4, V_5 represent the vertices of a directed graph. The adjacent matrix is given below: [4

	V_1	V_2	V_3	V_4	V_5
V_1	1	0	1	0	1
V_2	0	1	1	0	1
V_3	1	0	1	1	1
V_4	1	1	0	1	0
V_5	0	1	1	0	1

Construct a reachability matrix using breadth-first-search algorithm.

[Note: A reachability matrix contains value 1 if there exists a path between u and v otherwise 0; where u and v are any two vertex in the graph.]

4. a) What is hashing? How is it advantageous over sequential searching? Discuss about various collision resolution techniques available in hashing. [4

b) How to represent a polynomial using a linked list? Write a pseudo code to add two polynomials. [4

5. a) Write an algorithm to delete a node from a binary search tree. Find out time complexity of the algorithm. [4

b) What is a heap? Construct an max-heap with the following set of elements: [4

12, 23, 4, 15, 19, 33, 26, 42, 8, 40

Write an algorithm to sort the elements present in the heap.

6. a) Write an algorithm to construct an expression tree from a given postfix expression. Explain the algorithm with the following postfix expression. List the applications of the expression tree. [4

cab-*egh/-+

b) The degree of a node is the number of children it has. Prove by induction that in any binary tree, the numbers of leaves are one more than the number of nodes of degree 2. [4

7. a) Why should we prefer to represent a graph data structure using linked list? Explain the representation with a suitable example. [4

b) What is an AVL tree? Construct an AVL tree with the following set of numbers. [4

24, 13, 8, 20, 22, 36, 21, 17, 15, 14

8. Write short notes on <u>any two</u> [4 × 2

a) Algorithm Efficiency using logarithmic and linear logarithmic loop

b) Quick sort

c) Priority Queues

x x x x x