

Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 91561

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2014.

Third Semester

Information Technology

IT 2201/IT 33/080250005/10144 IT 304 — DATA STRUCTURES AND
ALGORITHMS

(Regulation 2008/2010)

Time : Three hours

Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. Define ADT
2. List four applications of stack.
3. Create the structure of a node in a Binary Search Tree.
4. What is the need for AVL trees?
5. What are the qualities of a good hashing function?
6. What do you mean by an equivalence relation?
7. Mention the applications of Depth First Traversal.
8. What is the suitable storage structure for developing algorithms for Minimum Spanning Trees? Why?
9. What are the features of divide and conquer algorithms.
10. What are randomized algorithms?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Apply suitable data structure and write an algorithm to match the left and right parentheses. Your algorithm should give the position in which matching / no matching brace occurs. For example, the string (a*(b+c)+d)) has left parenthesis at position 0, right parenthesis at position 7, there is no matching left parenthesis for the right brace at position 11. (12)
- (ii) Describe the significance of circular queue implementation for a particular application of your choice. (4)

Or

- (b) Write an algorithm that reads a list of integers from the keyboard, creates a linked list out of them, and prints the result. (16)

12. (a) Create a binary heap for the values 6,4,8,2,1,9,3,7,5 using a Minheap. Also provide algorithm for insertion and deletion. Indicate the results after deletion. (16)

Or

- (b) Write the algorithm for the following operations on Binary Search Tree
(i) insertion (ii) deletion (iii) findmin (iv) findmax. (16)
13. (a) Write the algorithm for performing the insertion and deletion operation into an initially empty extendible hashing data structure and illustrate with simple example. (16)

Or

- (b) (i) Explain the Disjoint set ADT with suitable examples. (8)
(ii) Write a note on the Path compression. (8)
14. (a) Explain the Prim's algorithm with an example. (16)

Or

- (b) (i) Write the Kruskal's algorithm for minimum spanning tree and illustrate with an example. (10)
(ii) Write the pseudocode for topological sorting for a directed graph. (6)
15. (a) Use Backtracking approach to solve the Knapsack problem and illustrate with simple example.

Or

- (b) (i) Explain the concepts in dynamic programming. (8)
(ii) Write a short note on NP complete problems. (8)