Reg. No. : | | | | | | | | | | |

Question Paper Code: 34802

B.E. / B.Tech. DEGREE EXAMINATION, APRIL 2019

Fourth Semester

Information Technology

01UIT402 - ANALYSIS AND DESIGN OF ALGORITHMS

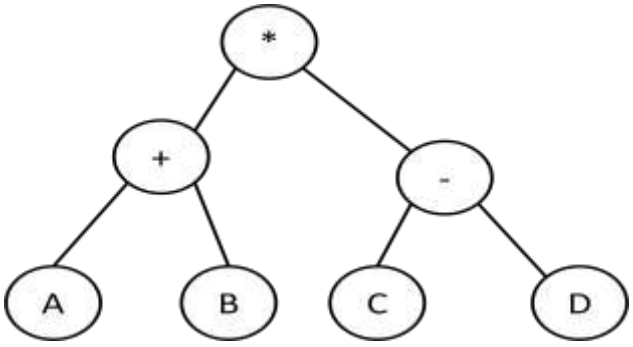
(Regulation 2013)

Duration: Three hours Maximum: 100 Marks

Answer ALL Questions.

PART A - (10 x 2 = 20 Marks)

1. Define Big – O notation.
2. Find gcd (31415, 14142) by applying Euclid's algorithm.
3. List the general plan for analyzing the time efficiency of non – recursive algorithms.
4. Define algorithm visualization.
5. How divide and conquer strategy generally decreases the time complexity?
6. Show all binary search tree traversal of the given expression tree.



7. List the important properties of heaps.
8. What is a Huffman code and tree?.

9. State subset sum problem.

10. Define NP Hard and NP Completeness.

PART - B (5 x 16 = 80 Marks)

11. (a) What is an algorithms? With a neat diagram, explain the various stages of algorithm design and analysis process. (16)

Or

(b) Explain all asymptotic notations used in algorithm analysis. (16)

12. (a) What is the mathematical analysis of recursive algorithms? Explain about the tower of Hanoi problem. (16)

Or

(b) Write a non-recursive algorithm to find whether the elements in a array are unique. Also analyze its efficiency. (16)

13. (a) What are the differences between DFS and BFS? Solve topological sorting problem using DFS algorithm with an example. (16)

Or

(b) Give a suitable example and explain the depth first search algorithm. (16)

14. (a) (i) Define Heap. Explain the properties of Heap. (8)

(ii) With a simple example, explain heap sort algorithm. (8)

Or

(b) Define the three variations of transform and conquer algorithms. Construct an AVL tree for the list 5, 6, 8, 3, 2, 4, 7 by successive insertions. State four rotation types used in the construction of the AVL tree and explain the same. (16)

15. (a) (i) How does backtracking work on the N Queens problem with an example? (8)

(ii) What is Hamiltonian circuit problem? Explain with an example using backtracking. (8)

Or

(b) Solve the given knapsack problem using greedy technique, n = 3, m = 20, $(p1, p2, p3)$ = (25, 24, 15), $(w1, w2, w3)$ = (18, 15, 10) and analyze algorithm time complexity. (16)