

FUTURE VISION BIE

One Stop for All Study Materials
& Lab Programs



Future Vision

By K B Hemanth Raj

Scan the QR Code to Visit the Web Page



Or

Visit : <https://hemanthrajhemu.github.io>

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering
& MORE...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/hemanthraj_hemu/

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: <https://bit.ly/FVBIESHARE>

| DESIGN AND ANALYSIS OF ALGORITHMS (Effective from the academic year 2018 -2019) SEMESTER – IV | | | |
|---|---------------|-------------------|----------------------|
| Course Code | 18CS42 | CIE Marks | 40 |
| Number of Contact Hours/Week | 3:2:0 | SEE Marks | 60 |
| Total Number of Contact Hours | 50 | Exam Hours | 03 |
| CREDITS –4 | | | |
| Course Learning Objectives: This course (18CS42) will enable students to: | | | |
| <ul style="list-style-type: none"> • Explain various computational problem solving techniques. • Apply appropriate method to solve a given problem. • Describe various methods of algorithm analysis. | | | |
| Module 1 | | | Contact Hours |
| Introduction: What is an Algorithm? (T2:1.1), Algorithm Specification (T2:1.2), Analysis Framework (T1:2.1), Performance Analysis: Space complexity, Time complexity (T2:1.3). Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ), and Little-oh notation (o), Mathematical analysis of Non-Recursive and recursive Algorithms with Examples (T1:2.2, 2.3, 2.4). Important Problem Types: Sorting, Searching, String processing, Graph Problems, Combinatorial Problems. Fundamental Data Structures: Stacks, Queues, Graphs, Trees, Sets and Dictionaries. (T1:1.3,1.4). RBT: L1, L2, L3 | | | 10 |
| Module 2 | | | |
| Divide and Conquer: General method, Binary search, Recurrence equation for divide and conquer, Finding the maximum and minimum (T2:3.1, 3.3, 3.4), Merge sort, Quick sort (T1:4.1, 4.2), Strassen’s matrix multiplication (T2:3.8), Advantages and Disadvantages of divide and conquer. Decrease and Conquer Approach: Topological Sort. (T1:5.3). RBT: L1, L2, L3 | | | 10 |
| Module 3 | | | |
| Greedy Method: General method, Coin Change Problem, Knapsack Problem, Job sequencing with deadlines (T2:4.1, 4.3, 4.5). Minimum cost spanning trees: Prim’s Algorithm, Kruskal’s Algorithm (T1:9.1, 9.2). Single source shortest paths: Dijkstra’s Algorithm (T1:9.3). Optimal Tree problem: Huffman Trees and Codes (T1:9.4). Transform and Conquer Approach: Heaps and Heap Sort (T1:6.4). RBT: L1, L2, L3 | | | 10 |
| Module 4 | | | |
| Dynamic Programming: General method with Examples, Multistage Graphs (T2:5.1, 5.2). Transitive Closure: Warshall’s Algorithm, All Pairs Shortest Paths: Floyd’s Algorithm, Optimal Binary Search Trees, Knapsack problem ((T1:8.2, 8.3, 8.4), Bellman-Ford Algorithm (T2:5.4), Travelling Sales Person problem (T2:5.9), Reliability design (T2:5.8). RBT: L1, L2, L3 | | | 10 |
| Module 5 | | | |
| Backtracking: General method (T2:7.1), N-Queens problem (T1:12.1), Sum of subsets problem (T1:12.1), Graph coloring (T2:7.4), Hamiltonian cycles (T2:7.5). Programme and Bound: Assignment Problem, Travelling Sales Person problem (T1:12.2), 0/1 Knapsack problem (T2:8.2, T1:12.2): LC Programme and Bound solution (T2:8.2), FIFO Programme and Bound solution (T2:8.2). NP-Complete and NP-Hard problems: Basic concepts, non- | | | 10 |

| | |
|---|--|
| deterministic algorithms, P, NP, NP-Complete, and NP-Hard classes (T2:11.1). | |
| RBT: L1, L2, L3 | |
| Course Outcomes: The student will be able to : | |
| <ul style="list-style-type: none"> • Describe computational solution to well known problems like searching, sorting etc. • Estimate the computational complexity of different algorithms. • Devise an algorithm using appropriate design strategies for problem solving. | |
| Question Paper Pattern: | |
| <ul style="list-style-type: none"> • The question paper will have ten questions. • Each full Question consisting of 20 marks • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module. | |
| Textbooks: | |
| <ol style="list-style-type: none"> 1. Introduction to the Design and Analysis of Algorithms, Anany Levitin., 2rd Edition, 2009. Pearson. 2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014, Universities Press | |
| Reference Books: | |
| <ol style="list-style-type: none"> 1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI. 2. Design and Analysis of Algorithms , S. Sridhar, Oxford (Higher Education). | |