

UE18CS280: DESIGN AND ANALYSIS OF ALGORITHMS

Course Objectives:

The objective(s) of this course is to,

- Learn to design and analyze algorithms with an emphasis on the resource utilization in terms of time and space.
- Learn various techniques in development of algorithms so that the effect of problem size and architecture design on the efficiency of the algorithm is appreciated.
- Learn to prove the correctness of algorithms.

Course Outcomes:

At the end of the course, the student will be able to:

- Identify the design technique used in an algorithm.
- Design an algorithm for a problem in a known design technique.
- Prove the correctness of an algorithm.
- Analyze the resource utilization of an algorithm in terms of time and space.
- Understand the limits of algorithms and the ways to cope with the limitations.

Course Content:

1. **Introduction:** Algorithms, Fundamentals of Algorithmic Problem Solving, Important Problem Types. **Analysis of Algorithm Efficiency:** Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non - Recursive and Recursive Algorithms.
2. **Brute Force:** Sequential Search, Brute Force String Matching, Selection Sort, Bubble Sort, Depth-First Search and Breadth-First Search, Exhaustive Search. **Divide-and-Conquer:** Merge Sort, Quick Sort, Binary Search, Binary Tree Traversals, Multiplication of Large Integers, Strassen's Matrix Multiplication and Master Theorem.
3. **Decrease-and-Conquer:** Insertion Sort, Topological Sorting, Algorithms for Generating Combinatorial Objects, Decrease-by-a-Constant-Factor Algorithms. **Transform-and-Conquer:** Presorting, Heap Sort, AVL Trees, Red-Black Trees, 2-3 Trees and B Trees.
4. **Space and Time Tradeoffs:** Sorting by Counting, Input Enhancement in String Matching - Horspool's and Boyer-Moore Algorithms. **Dynamic Programming:** Computing a Binomial Coefficient, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms.
5. **Greedy Technique:** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees. **Limitations of Algorithm Power:** Lower-Bound Arguments, Decision Trees, P, NP, and NP-Complete Problems. **Coping with the Limitations of Algorithm Power:** Backtracking, Branch-and-Bound.

Pre-requisite Courses: Problem Solving with C.

Reference Book(s):

1. "Introduction to the Design and Analysis of Algorithms", Anany Levitin, 2nd Edition, Pearson Education, 2011 (Updated Version of the Book).

2. "Introduction to Algorithms", Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, 3rd Edition, Prentice-Hall India, 2009.
3. "Fundamentals of Computer Algorithms", Horowitz, Sahni, Rajasekaran, 2/e, Universities Press, 2007.
4. "Algorithm Design", Jon Kleinberg, Eva Tardos, Pearson Education, 2006.