CSE325: Design and Analysis of Algorithms

Programme: B.Tech (All Branches) Year : IInd Semester : IVth
Course: CORE Credits : 3 Hours : 40

Course Context and Overview):

The objective of this course is to study paradigms and approaches used to analyze and design algorithms and to appreciate the impact of algorithm design in practice. This course also ensures that students understand how the worst-case, average-case time complexity of an algorithm is defined. During the course asymptotic notation will be used to provide a rough classification of algorithms. How different algorithms for fundamental problems in computer science and engineering work and compares with one another. The last part course will going to cover some fundamental problems in engineering for which it is unknown whether there exist efficient algorithms.

Prerequisites Courses:

CSE215: Data Structures

Course outcomes(COs):

On completion of this course, the students will have the ability to:
CO1: Analyze the asymptotic performance of algorithms.
CO2; Demonstrate a familiarity with major algorithms and data structures
CO3: Apply important algorithmic design paradigms and methods of analysis
CO4: Synthesize efficient algorithms in common engineering design situations

Course Topics:

Contents	**Lecture Hours**	
UNIT – 1 **Introduction**		8
1. What is algorithm?, Why analyze algorithms?, RAM Model of Computation.	2	
1.2 Asymptotic **notation:** big-Oh, big-Omega, big-Theta, little-oh	2	
2. Recurrence relations: iterative method, substitution method, recursion-tree method, Master method.	3	

UNIT –2 **Sorting**		6
1. Insertion-Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort	2	
2.2 Priority Queue , Heap data structure, Heap-sort	2	
2.3 Lower bound of sorting, Counting Sort, Radix Sort	2	
3. **UNIT-3** **Design Paradigm**		11
3.1 **Divide and Conquer :** Integer multiplication, Tromino Tiling, Strassen's Matrix multiplication, Counting inversions, Closet Pair of point in 2 Dimensions, Linear time selection algorithm	3	
3.2 **Greedy Algorithms:** Interval scheduling, Huffman Coding, Fractional Knapsack	3	
3.3 **Dynamic Programming:** Weighted interval scheduling, 0-1 Knapsack Problem, Matrix- chain multiplication, Longest Common subsequence	5	
UNIT-4 **Graph Algorithms**		12
4.1 **Graph Algorithms:** adjacency list and adjacency matrix representations of graphs	1	
4.2 **Breadth First Search(BFS)**: Applications of BFS like Shortest Path on un-weighted graph ,to check whether a graph is bipartite or not	2	
4.3 **Depth First Search(DFS):** Application Of DFS like Topological Sort, Cycle Detection, Checking Whether a Digraph is Strongly connected or not	3	
4.4 **Minimum Spanning Tree:** Kruskal's Algorithm and its using Union Find Data structure, Prim's Algorithm and its implementation using heap data structure	3	
4.5 **Shortest Path :** Bellman ford algorithm, Dijkstra's Algorithm, Application of Shortest path algorithms	3	

4. **UNIT-5** **NP-Completeness and Computational Intractability**		4
5.1 NP-Completeness-I -Motivation ,Definitions of NP, NP-Complete, NP hard Problems	2	
5.2 Polynomial-time reductions of known NP-Complete Problems	2	

Textbook references (IEEE format):

Text Book:

1.Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms* (Vol. 2, pp. 531-549). Cambridge: MIT press.

Reference books:

2. Skiena Steven, S. (2008). The algorithm design manual..

3. Kleinberg, J., & Tardos, É. (2006). *Algorithm design*. Pearson Education India

Evaluation Methods:

Item	**Weightage**
Assignment 1	10
Assignment 2	
Assignment 3	
Assignment 4	
Assignment 5	
2 Quizzes	25
Midterm	25
End Term Examination	40

Prepared By: Dinesh Khandelwal
Last Update: Jan, 2015

Evaluation Methods updated by: Sakthi Balan Muthiah
Last Update: Feb 10, 2017