| Course code | Course Name | L-T-P - Credits | Year of Introduction |
|---|---|---|---|
| **CS302** | **Design and Analysis of Algorithms** | **3-1-0-4** | **2016** |

**Prerequisite: Nil**

**Course Objectives**

- To introduce the concepts of Algorithm Analysis, Time Complexity, Space Complexity.
- To discuss various Algorithm Design Strategies with proper illustrative examples.
- To introduce Complexity Theory.

**Syllabus**

Introduction to Algorithm Analysis, Notions of Time and Space Complexity, Asymptotic Notations, Recurrence Equations and their solutions, Master's Theorem, Divide and Conquer and illustrative examples, AVL trees, Red-Black Trees, Union-find algorithms, Graph algorithms, Divide and Conquer, Dynamic Programming, Greedy Strategy, Back Tracking and Branch and Bound, Complexity classes

**Expected outcome**

The students will be able to

i. Analyze a given algorithm and express its time and space complexities in asymptotic notations.
ii. Solve recurrence equations using Iteration Method, Recurrence Tree Method and Master's Theorem.
iii. Design algorithms using Divide and Conquer Strategy.
iv. Compare Dynamic Programming and Divide and Conquer Strategies.
v. Solve Optimization problems using Greedy strategy.
vi. Design efficient algorithms using Back Tracking and Branch Bound Techniques for solving problems.
vii. Classify computational problems into P, NP, NP-Hard and NP-Complete.

**Text Books**


1. Ellis Horowitz, SartajSahni, SanguthevarRajasekaran, Computer Algorithms, Universities Press, 2007 [Modules 3,4,5]
2. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, MIT Press, 2009 [Modules 1,2,6]


**References**


1. Alfred V. Aho, John E. Hopcroft and Jeffrey D. Ullman, The Design and Analysis of Computer Algorithms, Pearson Education, 1999.
2. Anany Levitin, Introduction to the Design and Analysis of Algorithms, Pearson, 3rd Edition, 2011.
3. Gilles Brassard, Paul Bratley, Fundamentals of Algorithmics, Pearson Education, 1995.
4. Richard E. Neapolitan, Kumarss Naimipour, Foundations of Algorithms using C++ Psuedocode, Second Edition, 1997.


**Course Plan**

| Module | Contents | Hours | End Sem. Exam Marks |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **I** | ***Introduction to Algorithm Analysis***Time and Space Complexity- Elementary operations and Computation of Time Complexity- Best, worst and Average Case Complexities- Complexity Calculation of simple algorithms<br>***Recurrence Equations:***Solution of Recurrence Equations – Iteration Method and Recursion Tree Methods | **04**<br><br>**04** | **15 %** |
| **II** | ***Master's Theorem***(Proof not required) – examples, Asymptotic Notations and their properties- Application of Asymptotic Notations in Algorithm Analysis- Common Complexity Functions<br>***AVL Trees*** – rotations, Red-Black Trees insertion and deletion (Techniques only; algorithms not expected). B-Trees – insertion and deletion operations. Sets- Union and find operations on disjoint sets. | **05**<br><br>**05** | **15%** |
| **FIRST INTERNAL EXAM** | | | |
| **III** | ***Graphs*** – DFS and BFS traversals, complexity, Spanning trees – Minimum Cost Spanning Trees, single source shortest path algorithms, Topological sorting, strongly connected components. | **07** | **15%** |
| **IV** | ***Divide and Conquer:***The Control Abstraction, 2 way Merge sort, Strassen's Matrix Multiplication, Analysis<br>***Dynamic Programming :*** The control Abstraction- The Optimality Principle- Optimal matrix multiplication, Bellman-Ford Algorithm | **04**<br><br>**05** | **15%** |
| **SECOND INTERNAL EXAM** | | | |
| **V** | Analysis, Comparison of Divide and Conquer and Dynamic Programming strategies<br>***Greedy Strategy: -*** The Control Abstraction- the Fractional Knapsack Problem,<br>Minimal Cost Spanning Tree Computation- Prim's Algorithm – Kruskal's Algorithm. | **02**<br><br>**04**<br><br>**03** | **20%** |
| **VI** | ***Back Tracking: -***The Control Abstraction – The N Queen's Problem, 0/1 Knapsack Problem<br>***Branch and Bound:***Travelling Salesman Problem.<br>***Introduction to Complexity Theory :-***Tractable and Intractable Problems- The P and NP Classes- Polynomial Time Reductions - The NP- Hard and NP-Complete Classes | **03**<br>**03**<br><br>**03** | **20%** |
| **END SEMESTER EXAM** | | | |



**Question Paper Pattern**

1. There will be *five* parts in the question paper – A, B, C, D, E
2. Part A
   a. Total marks : 12
   b. *Four* questions each having *3* marks, uniformly covering modules I and II; All*four* questions have to be answered.
3. Part B
   a. Total marks : 18
   b. *Three* questions each having *9* marks, uniformly covering modules I and II; *Two* questions have to be answered. Each question can have a maximum of three subparts.
4. Part C

a. Total marks : 12
b. *Four* questions each having *3* marks, uniformly covering modules III and IV; All *four* questions have to be answered.

5. Part D
   a. Total marks : 18
   b. *Three* questions each having *9* marks, uniformly covering modules III and IV; *Two* questions have to be answered. Each question can have a maximum of three subparts
6. Part E
   a. Total Marks: 40
   b. *Six* questions each carrying 10 marks, uniformly covering modules V and VI; *four* questions have to be answered.
   c. A question can have a maximum of three sub-parts.

7. There should be at least 60% analytical/numerical questions.