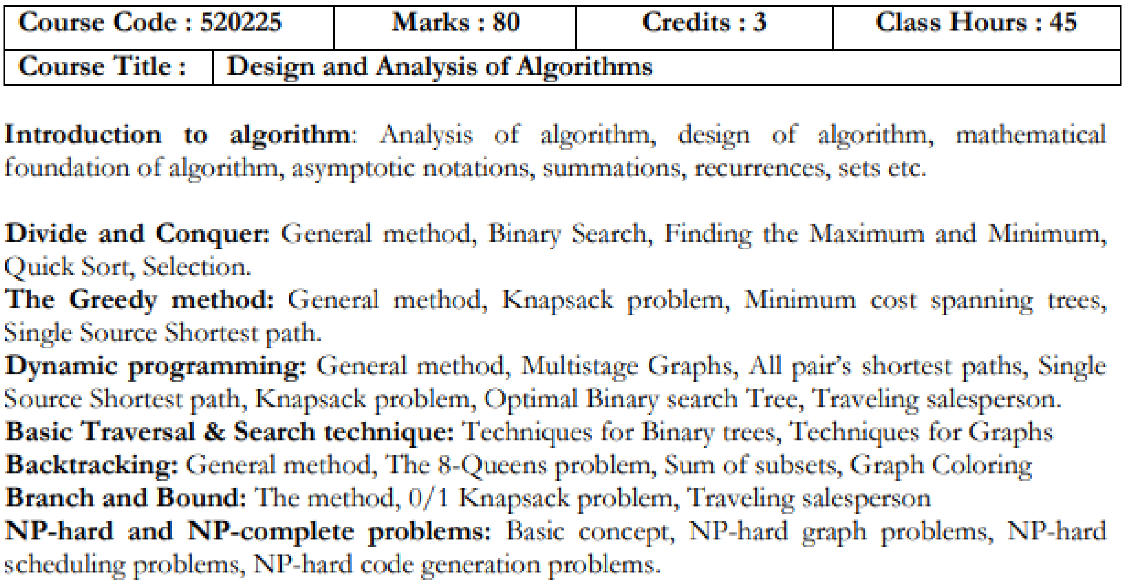
Design and Analysis of Algorithms



Course Code : 520225	Marks : 80	Credits : 3	Class Hours : 45
Course Title :	Design and Analysis of Algorithms		

Introduction to algorithm: Analysis of algorithm, design of algorithm, mathematical foundation of algorithm, asymptotic notations, summations, recurrences, sets etc.

Divide and Conquer: General method, Binary Search, Finding the Maximum and Minimum, Quick Sort, Selection.
The Greedy method: General method, Knapsack problem, Minimum cost spanning trees, Single Source Shortest path.
Dynamic programming: General method, Multistage Graphs, All pair's shortest paths, Single Source Shortest path, Knapsack problem, Optimal Binary search Tree, Traveling salesperson.
Basic Traversal & Search technique: Techniques for Binary trees, Techniques for Graphs
Backtracking: General method, The 8-Queens problem, Sum of subsets, Graph Coloring
Branch and Bound: The method, 0/1 Knapsack problem, Traveling salesperson
NP-hard and NP-complete problems: Basic concept, NP-hard graph problems, NP-hard scheduling problems, NP-hard code generation problems.

Chapter 1 (INTRODUCTION)

1. What is an algorithm? (2019, 2018,2016,2014,2009,208)
2. What do you mean by a computer algorithm (2015,2013)
3. How can you analyze an algorithm? (2018)
4. what are the criteria an algorithm must satisfy? (2013)
5. Why the need to study algorithms?

6. Why do we need to analyze algorithms?
7. What do you mean by algorithm validation? Explain how to validate an algorithm. (2019, 2009)
8. What is the difference between Algorithm and Pseudocode?
9. Define the concept of Algorithm design. (2017)
10. Write down the characteristics of an algorithm. (2019,2016,2008, 2014)
11. What are the various properties of an algorithm? (2009)
12. What do you mean by pseudocode?
13. Write down conventions for writing an algorithm. (2015,2013)
14. Mention the fundamental steps of solving an algorithm. (2013)
15. Write down the Pseudocode structure for looping and conditional statements.
16. Differentiate between Algorithm and Programming. (2017)
17. What is asymptotic notation? Describe different types of asymptotic notation with examples. (2018, 2017, 2015)
18. What is the growth function? With example, briefly explain each type of it. (2016)
19. Explain the time and space complexity of an algorithm. (2019, 2016,2009) Or, what do you mean by time complexity?
20. What do you mean by performance measurement?
21. How program performance can be evaluated? Explain with an example. (2016) (2008)
22. Which applications are not suitable for quick sorting and why? (2016)
23. Define the big "oh" (0) function. (2016)
24. Distinguish the best-case, average-case, and worst-case efficiency of an algorithm. (2013)
25. What is a randomized algorithm? Write down the advantages and disadvantages of the randomized algorithm. (2010)
26. Write down an algorithm for adding and deleting an item to end from a stack. (2018)
27. Write down a recursive algorithm to find the maximum and minimum number from a given set of numbers.
28. Write down the differences between the greedy method and dynamic programming. (2018)
29. Define the following terms: Tree edge; Back edge; Cross edge.
30. Write down the pseudo code conversions for writing an algorithm. (2014, 2011)
31. Define "Big oh', Big theta' and 'Big omega".
32. Explain how the time complexity of any program is computed.
33. Write down the application of the BFS and DFS algorithms. (2014, 2017)
34. Briefly explain the BFS algorithm with an example.
35. What is the reachability problem?
36. Write a non-deterministic algorithm for searching an element. (2019)

Chapter 2 (DIVIDE AND CONQUER)

1. What are the Divide and Conquer methods? Describe how they work. Can you give any examples of the types of problems where this approach might be used? (2019, 2016)

2. How does Divide and Conquer method work? (2019, 2018, 2017)
3. What is the difference between quick sort and merge sort? (2009, 2011)
4. Explain Hoare's method of partitioning with a suitable example.
5. Write an Algorithm to find the maximum and minimum value from an array. (2009, 2016)
6. What are the Divide and Conquer methods? (2019, 2009, 2011, 2012)
7. Write down the control abstraction for the divide and conquer strategy. (2008, 2012)
8. Write the recursive and iterative process of the binary search algorithm. (2019, 2016)
9. Write down an algorithm for binary search.
10. Write an Algorithm to find the maximum and minimum value from a number list using the divide and conquer method. (2008)
11. Define the technique of merging. Write down the merge sort algorithm. (2016)
12. How does a binary search work?
13. Write down the advantages and disadvantages of binary search over linear search algorithm.
14. What is a subset problem? (2017, 2017)
15. What are the general plans for divide-and-conquer algorithms? (2013, 2011)

Chapter 3 (THE GREEDY METHOD)

1. What is greedy choice property? (2016, 2011)
2. Describe the steps required to develop a greedy algorithm. (2010, 2011)
3. What are the different types of problems that can be solved using the greedy method? (2010)
4. Write down the major Areas of Application of the Greedy Algorithms.
5. Describe the general approach of the greedy method using control abstraction. (2018)
6. What is control abstraction? (2018)
7. Define a spanning tree. (2019, 2018, 2008)
8. Write the application of the spanning tree. (2013)
9. Describe the Components of the Greedy Algorithm.
10. List the important properties of the greedy method. (2010)
11. Describe Prim's algorithm for finding a minimum cost-spanning tree with a suitable example. (2008)
12. Write down the difference between the greedy method and dynamic programming. (2016, 2008)
13. Differentiate optimal solution and feasible solution.
14. What is the knapsack problem? (2019, 2008)
15. Write down the algorithm for greedy strategies for the knapsack problem. (2008)
16. Explain the multistage graph corresponding to the backward approach with a suitable example.
17. Write Krushal's algorithm to construct a minimum cost-spanning tree.
18. Write down the control abstraction of the greedy method.
19. Explain the greedy fractional knapsack algorithm with a suitable example.
20. How to create a Greedy Algorithm?

Chapter 4 (DYNAMIC PROGRAMMING)

1. What do you mean by dynamic programming? (2010)
2. What is the difference between an optimal solution and a feasible solution? (2010)
3. Define A Binary Search Tree (BST)
4. Describe OBST as dynamic programming with an example.
5. Define binary heap tree. (2018)
6. Define the Max-heap tree and Min-heap Tree.
7. Define the bubble sort algorithm with an example.
8. Define insertion sort with an example.
9. Define selection sort with an example.
10. Write down the Steps of the Dynamic Programming Approach.
11. Write down the Applications of the Dynamic Programming Approach.
12. Define multistage graphs with examples.
13. Define the Travelling salesman problem with an example.
14. In the following example, find the steps to solve the traveling salesman problem.
15. Write down the quick sort algorithm.
16. Define a quick sort algorithm with advantages and disadvantages.
17. Which applications are not suitable for quick short algorithms and why? (2017)
18. Define NP-Completeness. (2018)
19. Prove that TSP is NP-Complete.
20. Write down the bellman-Ford algorithm.
21. List the important futures of dynamic programming.
22. Define implicit and explicit constraints. (2018, 2014)
23. Write down Dijkstra's algorithm for the single source shortest path. (2019, 2013)

Chapter 5 (BASIC TRAVERSAL AND SEARCH TECHNIQUES)

1. Define linear search and binary search.
2. Define binary tree and skewed binary tree with an example. (2011, 2008)
3. Discuss the different traversal techniques of a binary tree. (2014, 2011,2008)
4. What are the properties of a Minimum Spanning Tree (MST)? (2017)
5. How to solve the single source shortest paths problem? (2019, 2017)
6. What is the difference between linear search and binary search? (2019, 2011)
7. Write down the advantages and disadvantages of the BFS and DFS traversal algorithms. (2011)
8. Write down the DFS traversal algorithm.
9. Write down the BFS traversal algorithm.
10. Write down the difference between BFS and DFS.
11. Discuss optimal binary search tree problems. (2018, 2011)
12. Prove that the best case complexity of quick sort is O(n log n). (2016)
13. What is an Optimal Binary Search Tree (OBST)? (2016)

14. Write down the algorithm for recursive binary search.
15. Show that the average case complexity of the quick sort technique is 0 (nlogn). (2015)
16. Write down pseudo code for breadth-first search. (2019, 2014, 2011)

Chapter 6 (BACKTRACKING)

1. What do you mean by backtracking?
2. Define the Hamilton Cycle, Chromatic-number, State space tree, and Bounding function. (2014)
3. Write down the control abstraction of LC search.
4. Prove that BPS (Breadth First Search) visits all vertices reachable from V.
5. What do you mean by the shortest path problem? (2011)
6. Write Dijkstra's algorithm for single sources to shorten the path problem. (2019, 2011, 2018)
7. Write down the principle of optimality.
8. Define implicit and explicit constraints.
9. What are the factors that influence the efficiency of the backtracking algorithm? (2018,2015,2011)
10. Write an algorithm for the n-Queens problem. (2015,2010)
11. Write an algorithm for the sum of the subset problem
12. White short notes on 15 puzzle problems. (2016)
13. Write down the iterative and recursive backtracking algorithm. (2018, 2010)
14. What are the 4 queen problems?
15. Explain the multistage graph corresponding to the backward approach with a suitable Example.
16. Write the derived facts for solving the procedure.
17. Discuss the 8 Queens problem. (2010, 2009)

Chapter 7 (BRANCH AND BOUND)

1. What do you mean by the branch-bound method?
2. Discuss the 8 Queen problems.
3. What is all pair's shortest path algorithm?
4. Write down the algorithm for all pair's shortest paths.
5. Show at least two solutions for the 8-queen problem.
6. What is graph coloring and the way, it is used?
7. Differentiate backtracking and branch-bound method. (2019, 2018, 2008, 2010, 2011)
8. Write down the m-coloring algorithm. (2018)
9. Differentiate between Branch and Bound over the backtracking algorithm. (2016)
10. When can a path be terminated in a branch and bound algorithm? (2015)
11. Use a recursion tree to determine a good asymptotic upper bound on the recurrence.T(n) = 2T(n/2)+n2