QUESTION BANK
Design and Analysis of Algorithm

UNIT I - ALGORITHM ANALYSIS

1. Define algorithm.
2. What is big 'Oh' notation?
3. State the best case and worst case analysis for linear search.
4. Solve the recurrence equation.

   T (n) =2T (n-1) +n2^n+n2.
5. Give the recurrence equation for the worst case behavior of merge sort.
6. What is the average case complexity of linear search algorithm?
7. A) Define the asymptotic notations used for best case average case and worst case Analysis of algorithm.

   B) Write an algorithm for finding maximum element of an array; perform best and Average case complexity with appropriate order notations.
8. Write an algorithm to find mean and variance of an array perform best, worst and
9. Average case complexity, defining the notations used for each type of analysis.
10. Derive the recurrence relation for Fibonacci series; perform complexity analysis for the same.
11. Explain the various asymptotic notations with the properties.
12. Explain linear search with example

UNIT II-DIVIDE AND CONQUER

1. Define Substitution method.
2. Analysis the various cases of complexity for Binary search.
3. Define Knapsack problem.
4. Write the complexity of divide and Conquer algorithms.
5. Sort the numbers using merge sort.
6. List out the disadvantage of merge sort.

7. What are the four feasible solutions for

n=3,m=20,(p1,p2,p3)=(25,24,15),(w1,w2,w3)=18,15,10).

8. Give the time efficiency and drawback of merge sort algorithm.

9. Write a pseudo code for a divide and conquer algorithm for finding the position of the largest element in an array of N numbers.

10. What is recursive call?

11. List out any two drawbacks of binary search algorithm.

12. Write the procedure for selection sort.

13. Explain the algorithm for maximum and minimum numbers in an array.

14. Give a detailed note on Divide and Conquer techniques.

15. Sort the following set of elements using merge sort

12, 24, 8, 71, 4, 23, 6, 89, 56

16. Write an algorithm for searching an element using Binary search Method. Give an example.

17. (A) Write a pseudo code for a divide and conquer algorithm for merging two sorted arrays into a single sorted one. Explain with an example.

(b) Setup and solve a recurrence relation for the number of key comparisons made by the above pseudo code.

18. A) Write an algorithm to sort a set of N numbers using insertion sort.

(b) Trace the algorithm for the following set of numbers.

20,35,18,8,14,41,3,39.

19. 7. Explain in detail merge sort. Illustrate the algorithm with a numeric example. Provide complete analysis of the same

UNIT III - DYNAMIC PROGRAMMING

1. Write any four examples for Brute Force Approach.

2. Define Dynamic programming.

3. Define multistage graph problem.

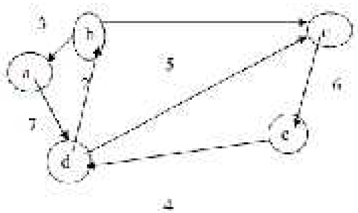
4. What is the difference between forward & backward approach?

5. Define all-pair shortest path problem.

6. Define Optimal Binary Search trees.
7. What is 0/1 Knapsack.
8. What is the procedure to solve traveling Salesman problem.
9. List out the advantages of Dynamic programming.
10. Solve the all-pairs shortest path problem for the digraph with the weight matrix given below.

	A	B	C	D
A	0	_	_	3
B	2	0	_	_
C	_	7	0	1
D	6	_	_	0

11. Solve the following instance of the single source shortest path problem with vertex ‘a’ as the source.



12. Explain the Multistage graph with example.
13. Explain the 0/1 knapsack with an algorithm.
14. Describe the Traveling salesman problem & discuss how to solve it using Dynamic Programming.

UNIT IV - BACKTRACKING

1. Define sum of subsets problem.
2. Define Backtracking.
3. What are the applications of backtracking?
4. What are the algorithm design techniques?

5. Define n-queens problem.
6. Define Hamiltonian Circuit problem in an undirected Graph.
7. What is state space tree?
8. State the principle of Backtracking.
9. State if Backtracking always produces optimal solution.

10. What is Backtracking? Explain in detail.
11. Explain Subset-sum Problem & Discuss the possible solution strategies using backtracking.
12. Discuss the use of greedy method in solving knapsack problem and subset sum problem.
13. Write short notes on
    (a) Graph coloring (8)
    (b) 8-Queens problem (8)
14. Apply Backtracking technique to solve the following instance of the subset sum problems. s= (1, 3, 4, 5) & d=11 (16)
15. Explain subset-sum problem and discuss the possible solution strategies using backtracking.
16. Explain 8-Queens problem with an algorithm. Explain why backtracking is defined as a default procedure of last resort for solving problems.
17. Using Backtracking enumerate how can you solve the following problems
    (a) 8-queens problem
    (b) Hamiltonian circuit problem

UNIT V - TRAVERSALS, BRANCH AND BOUND

1. Define Traversal of Trees.
2. What are the different ways of traversal of Trees?
3. Define connected components.
4. When do you say a tree as minimum spanning tree?

5. What is a minimum cost spanning tree?
6. Define depth first searching techniques.
7. Define bi connected components.
8. Compare Backtracking & Branch and Bound techniques with an example.
9. What are the applications of branch & bound?
10. Define Nondeterministic algorithms.
11. Define Deterministic algorithms.
12. What are the three function specify for Nondeterministic algorithm.
13. Compare NP-hard and NP-completeness.
14. Explain NP-hard and NP complete problems with example.
15. Explain connected components and bi-connected components with pseudo code.
16. What is branch and bound? Explain detail.
17. Discuss the solution for knapsack problem using branch bound techniques.