

DESIGN AND ANALYSIS OF ALGORITHMS QUESTION BANK

I-UNIT PART-A

1. Define Algorithm & Notion of algorithm.
2. What is analysis framework?
3. What are the algorithm design techniques?
4. How is an algorithm's time efficiency measured?
5. Mention any four classes of algorithm efficiency.
6. Define Order of Growth.
7. State the following Terms.
 - (i) Time Complexity
 - (ii) Space Complexity
8. What are the various asymptotic Notations?
9. What are the important problem types?
10. Define algorithmic Strategy (or) Algorithmic Technique.
11. What are the various algorithm strategies (or) algorithm Techniques?
12. What are the ways to specify an algorithm?
13. Define Best case Time Complexity .
14. Define Worst case Time Complexity.
15. Define Average case time complexity.
16. What are the Basic Efficiency Classes.
17. Define Asymptotic Notation.
18. How to calculate the GCD value?

PART-B:

1. (a) Describe the steps in analyzing & coding an algorithm. (10)
(b) Explain some of the problem types used in the design of algorithm. (6)
2. (a) Discuss the fundamentals of analysis framework . (10)
(b) Explain the various asymptotic notations used in algorithm design. (6)
3. (a) Explain the general framework for analyzing the efficiency

of algorithm. (8)

(b) Explain the various Asymptotic efficiencies of an algorithm. (8)

4. (a) Explain the basic efficiency classes. (10)

(b) Explain briefly the concept of algorithmic strategies. (6)

5. Describe briefly the notions of complexity of an algorithm. (16)

6. (a) What is Pseudo-code? Explain with an example. (8)

(b) Find the complexity $C(n)$ of the algorithm for the worst case, best case and average case. (Evaluate average case complexity for $n=3$, Where n is the number of inputs) (8)

II-UNIT

PART-A

1. What is algorithm visualization?

2. What is algorithm animation?

3. What is the tool for analyzing the time efficiency of a non - recursive algorithm?

4. What are the difference between mathematical & Empirical analysis?

5. What are the two applications of algorithm visualization?

6. What are the three ways by which an algorithm can be analyzed?

7. How the mathematical analysis can be performed?

8. Give any four features of algorithmic visualization.

9. What are the two approaches of algorithmic visualization?

10. What are the two categories of algorithms?

11. What is the mathematical modeling?

PART-B

1. Write an algorithm for a given numbers n to generate the n^{th}

number of the Fibonacci sequence. (16)

2. Explain pros and cons of the empirical analysis of algorithm. (16)

3. (a) Explain the necessary steps for analyzing efficiency of recursive algorithms with an example. (10)

(b) What is empirical analysis of an algorithm? Discuss its

strength & Weakness. (6)

4. Show a tree structure for recursive calls made in the problem

of Tower of Hanoi. (16)

5. (a) Write short notes on algorithm visualization & applications. (10)

(b) Discuss the features of animation of an algorithm. (6)

6. (a) Design a non-recursive algorithm for computing the product

of two $n \times n$ matrices and also find the time efficiency of algorithm. (10)

(b) Give the general plan for empirical analysis. (6)

III-UNIT

PART-A

1. Write any four examples for Brute Force Approach.

2. Find the number of comparisons made by the sequential search in the worst & Best case.

3. Give the time efficiency & Drawback of merge sort Algorithm.

4. What is the difference between DFS & BFS?

5. What is the Brute Force Algorithmic Strategy?

6. State the time complexity of following:

(i) Bubble sort

(ii) Selection sort

(iii) Sequential search

(iv) Brute force string matching

7. What are the features of Brute force String matching algorithm?

8. Give any two strength & Weakness of Brute force algorithm.

9. Explain Brute force string matching algorithm.

10. Define "Divide & Conquer Technique".

11. State Master's Theorem.

12. Define Merge sort & explain three steps of Merge sort.

13. Define Quick sort & explain three steps of Quick sort.

14. Define Binary Search.

15. What are the applications of binary search?

16. State advantages & Disadvantages of binary search.

17. Explain Binary search tree.

18. What is the recurrence relation for divide & conquer?
19. Explain Decrease & Conquer.
20. What are the variations of Decrease & Conquer?
21. What are the applications of decrease by constant?
22. Write any four advantages of insertion sort.
23. What are the two types of searching algorithm?
24. Define the Following Terms:
 - (i) Tree Edge
 - (ii) Back Edge
 - (iii) Cross Edge

PART-B:

1. Explain Quick sort algorithm with suitable Example. (16)
2. (a) Write an algorithm to sort a set of 'N' numbers using insertion sort. (8)
(b) Explain the difference between depth first search & Breadth first search. (8)
3. (a) Write a pseudo code for divide & conquer algorithm for merging two sorted arrays into a single sorted one. Explain with example. (12)
(b) Set up & solve a recurrence relation for the number of key comparisons made by above pseudo code. (4)
4. Give a suitable example & explain the Breadth first search & Depth first search. (16)
5. Find the number of comparisons made by the sentinel version of sequential search algorithm for in,
 - (i) Worst case
 - (ii) Average case (16)
6. Design a recursive Decrease-by-one algorithm for sorting the n real numbers in any array with an example & also determine the number of key comparisons & time efficiency of the algorithm. (16)
7. (a) Give an algorithm for selection sort & analyze your algorithm. (10)

(b) Give Strength & Weakness of Brute force algorithm. (6)

IV-UNIT

PART-A

1. What is pre-sorting? Give examples.
2. How efficient is prim's algorithm?
3. Define concept of transform & conquer.
4. What are the three variations of transform & conquer?
5. State the following terms:
 - (i) Balanced Tree
 - (ii) Unbalanced Tree
6. What is height of balanced tree?
7. What is balance factor?
8. Define rotation & what are the four types of rotations?
9. What are the drawbacks of AVL trees?
10. What is 2-3 trees?
11. Define Heap. & what are the two types heap?
12. What are the important properties of heap?
13. Define complete binary tree.
14. What is the height of the tree?(or)depth of the tree?
15. Define almost complete binary tree.
16. How to construct a heap?
17. What are the features of heap sort?
18. Define heapsort.
19. State the properties of AVL trees.
20. What is dynamic programming?
21. Compare divide & conquer and Dynamic programming.
22. How the problems can be solved using dynamic programming?
23. Define Warshall's algorithm.
24. Define Floyd's algorithm.
25. Give any two properties of dynamic programming approach.
26. Define principle of optimality.
27. What is OBST?
28. What is Greedy method?
29. Compare Greedy algorithm & Dynamic programming.
30. Define Prim's algorithm.

31. Define Kruskal's algorithm.
32. Define Dijkstra's shortest path algorithm.
33. How to obtain Huffman's Code?
34. What are the applications of Huffman trees?
35. What are the applications of spanning trees?

PART-B

1. (a) Construct a minimum spanning tree using Kruskal's algorithm with your own example. (10)
(b) Properties of Heap sort. (6)
2. (a) What do you mean by Huffman code? (10)
(b) Explain single L-Rotation & of the double RL-Rotation with general form. (6)
3. Define AVL tree. Explain the construction sequence of AVL tree with simple example. (16)
4. Solve the all pair shortest path problem for the diagraph with the weighted matrix given below:-
a b c d
a 0 ∞ 3 ∞ (16)
b 2 0 ∞ ∞
c ∞ 7 0 1
d 6 ∞ ∞ 0
5. Explain Warshall's & Floyd's Algorithm. (16)
6. Define Spanning tree. Discuss design steps in Prim's algorithm to construct minimum spanning tree with an example. (16)
7. (a) Explain Kruskal's algorithm. (8)
(b) Explain Dijkstra algorithm (8)

V-UNIT

PART-A

1. Define Backtracking.
2. What are the applications of backtracking?
3. What are the algorithm design techniques?
4. Define n-queens problem.
5. Define Hamiltonian Circuit problem.
6. Define sum of subset problem.
7. What is state space tree?
8. Define Branch & Bound method.
9. Define assignment problem.

10. What is promising & non-promising node?
11. Define Knapsack's problem.
12. Define Travelling salesman problem.
13. State principle of backtracking.
14. Compare Backtracking & Branch and Bound techniques with an example.
15. What are the applications of branch & bound?(or) What are the examples of branch & bound?
16. In Backtracking method, how the problem can be categorized?
17. How should be determine the solution in backtracking algorithm?
18. Obtain all possible solutions to 4-Queen's problem.
19. Generate atleast 3-solutions for 5-Queen's problem.
20. Draw a pruned state space tree for a given sum of subset problem:
 $S = \{3, 4, 5, 6\}$ and $d = 13$