1. Define Algorithm. List and Explain characteristics of algorithm.
2. Explain why it is important to analyse algorithms before implementing them? Provide examples to illustrate your answer.
3. Discuss the factors for comparing algorithms
4. Explain the types of analysis done on an algorithm.
5. Analyze the performance characteristics of algorithms based on their rate of growth. How does the rate of growth impact the efficiency of algorithms?
6. Explain different properties of notations.
7. Define Tree. Provide a glossary of key terms associated with Trees.
8. Discuss the concept of AVL Trees and their role in maintaining balance in BSTs.
9. Analyze the following binary tree and find the Pre-order, In-order, Post order traversal.
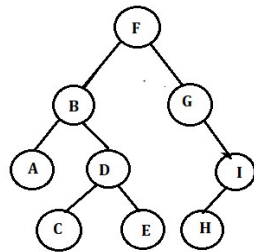
10. Explain Depth-First Search (DFS) and Breadth-First Search (BFS) algorithms with examples.
11. Calculate the minimum cost spanning tree using Prim's Algorithm for the following graph.
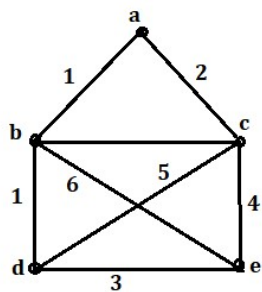

12. Evaluate following graph and find the shortest path using Dijkstra's Algorithm.
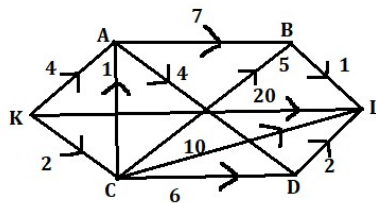
13. Write a short note on classification of algorithm by implementation method and design method.
14. What is Greedy Algorithm? Explain with example.
15. Explain Huffman coding with example.
16. What is Knapsack Problem? Explain with example. trees and their role in balancing binary search trees (BSTs).
17. List and explain the properties of Dynamic Programming Strategy.
18. Explain advantages and disadvantages of divide and conquer.
19. List and explain the commonly used asymptotic notations to calculate the running time complexity of an algorithm.
20. Write a short note on Dynamic Programming Strategy.

21. Explain Binary Tree Traversal with example.
22. Analyze the following graph and find the minimum spanning tree using Kruskal's and Prim's Algorithm.
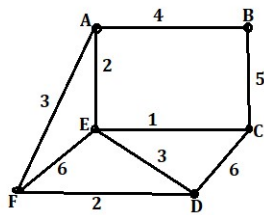
23. Write a short note on advantages and disadvantages of greedy method.
24. Explain the Master Theorem for Divide and Conquer.
25. Define asymptotic notation and discuss its significance in the analysis of algorithms. How does it help in comparing different algorithms?
26. Explain why it is important to analyze algorithms before implementing them. Provide examples to illustrate your answer.
27. Evaluate the properties of asymptotic notations and their role in algorithm analysis.
28. Compare the Big-O, Omega-Ω, and Theta-Θ notations. When would you use each notation in algorithm analysis?
29. Explain the Master Theorem for Divide and Conquer algorithms.
30. What are the characteristics an algorithm must have explain in brief.
31. Explain different types of binary tree.
32. Consider following graph and explain the working of topological sort.
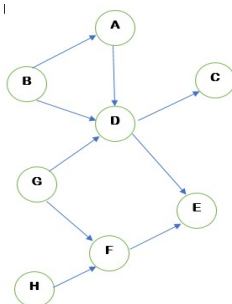
33. Write a short note on DFS and BFS.
34. Explain different types of binary tree.
35. Explain pre-order, in-order and post-order traversal with example.
36. Find the minimum cost spanning tree using Kruskal's Algorithm for the following graph.
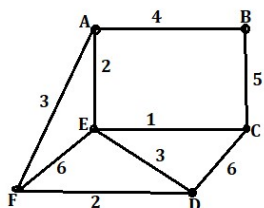

37. Describe classifications of algorithm in detail.
38. Explain Greedy Algorithm with example.
39. Write a short note on Divide and Conquer Strategy.

40. Consider the capacity of the knapsack is M=20 and the objects are as shown in the table. Apply greedy approach to solve this problem to gain maximum profit.

Objects	Profit	Weight
O1	25	18
O2	24	15
O2	15	10

41. What is Dynamic Programming Strategy? Explain properties of Dynamic Programming Strategy.
42. Explain the longest common subsequence problem.
43. Compare the performance of two algorithms using asymptotic analysis. Provide a detailed explanation of how you would determine which algorithm is more efficient.
44. Describe AVL Tree in detail.
45. Discuss the significance of common logarithms and summations in algorithm analysis. Provide examples of how they are used to analyze algorithms.
46. Evaluate following graph and find the shortest path using Dijkstra's Algorithm.
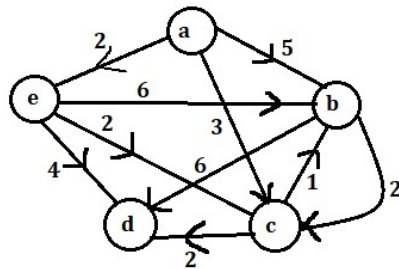
47. Explain Huffman coding using the following character frequencies:
48. Message M = 100 characters
    A=50 B=10 C=30 D=5 E=3 F=2
49. Discuss the factors for comparing algorithms.
50. Define asymptotic notation and discuss its significance in the analysis of algorithms. How does it help in comparing different algorithms?
51. Explain the terms in Rate of growth.
52. Use Master's theorem to solve the recurrence relation shown below-
53. T(n) = 3T(n/2) + n2
54. T(n)=2T(n/2) +n
55. Describe Big-O and Omega Ω in algorithm analysis.
56. Explain the Master Theorem for Divide and Conquer algorithms.
57. Define an algorithm. State its essential characteristics. Develop an algorithm for calculating factorial of an integer the user provided.
58. Define the Following terms of Tree:
59. Path 2. Root 3. Parent 4. Depth of tree 5. Degree of node.
60. Explain the working of topological sort considering following graph
61. What are graphs? Enlist Types of Graphs. And State Applications, Advantages and Disadvantages of Graph.
62. State and Explain types of binary tree traversal. Demonstrate with suitable example
63. Find the minimum cost spanning tree using Kruskal's Algorithm for the following graph.
64. State and Explain types of Graph traversal technique.
65. Enlist the Algorithms Classification methods. Discuss classification   by implementation method in detail.

66. What is Greedy Algorithm? Explain Advantages and disadvantages of Greedy Algorithm. State its applications.
67. Explain any 5 applications of divide and conquer strategy.
68. Calculate how many bits are saved using Huffman Code of following Example:

Character	Frequency
A	12
B	2
C	7
D	13
E	14
F	85

69. What is Dynamic Programming Strategy? Explain properties of Dynamic Programming Strategy.
70. Describe master theorem in detail.
71. Explain Guidelines for Asymptotic Analysis.
72. Describe Advantages and Disadvantages of Threaded Binary Tree.
73. Discuss the significance of common logarithms and summations in algorithm analysis. Provide examples of how they are used to analyze algorithms.
74. Consider following graph and find the shortest path using Dijkstra's shortest path algorithm using Prim's Algorithm.
75. Explain Fibonacci number using dynamic programming approach.
76. Distinguish Between a Dynamic Programming Algorithm and Recursion