**Unit 1: Introduction (6 hours)**

1. **Define an algorithm.** What are the key characteristics that make a good algorithm?
2. **Explain the importance of algorithm specification.** How does pseudocode help in this process?
3. **Discuss space complexity and time complexity.** Provide examples.
4. **What is asymptotic notation?** Explain with examples of Big O, Omega, and Theta notations.
5. **Describe the concept of randomized algorithms.** What are their advantages and disadvantages?

**Unit 2: Basic Data Structures (2 hours)**

1. **Explain the structure and operations of stacks and queues.** How are they used in algorithm design?
2. **What are binary trees?** Discuss their types and applications.
3. **Differentiate between binary search trees and heaps.** Provide use cases for each.
4. **Explain the concept of sets and disjoint set union.** How is it applied in graph algorithms?
5. **Discuss different ways to represent graphs.** What are the advantages of each representation?

**Unit 3: Divide and Conquer Method (5 hours)**

1. **What is the divide and conquer strategy?** Provide an example of an algorithm that uses this strategy.
2. **Describe the binary search algorithm.** How does it improve search efficiency?
3. **Explain the process of merge sort and quicksort.** Compare their time complexities.
4. **Discuss Strassen’s matrix multiplication algorithm.** How does it differ from the conventional method?

**Unit 4: The Greedy Method (7 hours)**

1. **Explain the greedy method.** What are its main advantages and limitations?
2. **What is the fractional knapsack problem?** How is it solved using the greedy approach?
3. **Discuss the job sequencing with deadlines problem.** How is it solved using the greedy method?
4. **Compare Prim's and Kruskal's algorithms.** How do they find minimum cost spanning trees?
5. **Explain the optimal merge pattern problem.** How does the greedy method solve it?

**Unit 5: Dynamic Programming (9 hours)**

1. **What is dynamic programming?** How does it differ from the greedy method?
2. **Explain the 0/1 knapsack problem.** How is it solved using dynamic programming?
3. **Discuss the multistage graph problem.** How does dynamic programming help in finding the shortest path?
4. **What is the Traveling Salesman Problem (TSP)?** Explain a dynamic programming approach to solve it.
5. **Define reliability design in the context of dynamic programming.** What are its applications?

**Unit 6: Basic Traversal and Search Techniques (7 hours)**

1. **Describe breadth-first search (BFS) and depth-first search (DFS).** How do they differ in traversing graphs?
2. **Explain the significance of connected components and spanning trees in graph theory.**
3. **What is the role of biconnected components in graph theory?** How is DFS used to identify them?

**Unit 7: Backtracking (5 hours)**

1. **What is backtracking?** Provide an example of a problem solved using this technique.
2. **Explain the 8-Queens problem.** How is it solved using backtracking?
3. **Discuss the sum of subsets problem.** How does backtracking help in finding the solution?
4. **What is the graph coloring problem?** How can backtracking be used to solve it?
5. **Describe the Hamiltonian cycle problem.** How is backtracking applied to find a solution?