Dynamic programming

1. **Fibonacci Sequence**
    - Find the nth Fibonacci number using dynamic programming (with memoization or tabulation).
2. **0/1 Knapsack Problem**
    - Given weights and values of items and a knapsack with a weight limit, find the maximum value you can put in the knapsack.
3. **Longest Increasing Subsequence (LIS)**
    - Given an array of integers, find the length of the longest strictly increasing subsequence.
4. **Coin Change Problem**
    - Given a set of coin denominations and an amount, find the minimum number of coins required to make the amount.
5. **Longest Common Subsequence (LCS)**
    - Given two strings, find the length of the longest subsequence common to both strings.
6. **Edit Distance (Levenshtein Distance)**
    - Given two strings, find the minimum number of operations (insertions, deletions, substitutions) required to convert one string into the other.
7. **Partition Problem**
    - Given a set of integers, determine if it's possible to partition the set into two subsets with equal sum.
8. **Subset Sum Problem**
    - Given a set of numbers, find whether there is a subset whose sum is equal to a given target value.
9. **Matrix Chain Multiplication**
    - Given a sequence of matrices, find the most efficient way to multiply them (minimizing the number of scalar multiplications).
10. **Rod Cutting Problem**
    - Given a rod of length n and a set of prices for different lengths of the rod, find the maximum profit obtainable by cutting the rod into smaller lengths.
11. **House Robber Problem**

- Given a list of non-negative integers representing the amount of money in each house, determine the maximum amount of money you can rob tonight without robbing two adjacent houses.

12. **Minimum Path Sum in a Grid**

- Given a 2D grid of non-negative integers, find a path from the top-left corner to the bottom-right corner, where you can only move down or right, and the sum of the numbers along the path is minimized.

13. **Word Break Problem**

- Given a string and a dictionary of words, determine if the string can be segmented into a space-separated sequence of dictionary words.

14. **Maximum Subarray Sum (Kadane's Algorithm)**

- Given an integer array, find the contiguous subarray (containing at least one number) which has the largest sum and return that sum.

15. **Unique Paths**

- Given a grid with dimensions `m x n`, find the number of unique paths from the top-left corner to the bottom-right corner, moving only down or right.

Graph

1. **Find the Shortest Path in an Unweighted Graph**
   - **Problem:** Given an unweighted graph (either directed or undirected) and a source node, find the shortest path to all other nodes using BFS.
2. **Connected Components in an Undirected Graph**
   - **Problem:** Given an undirected graph, find all the connected components using BFS.
3. **Level of Each Node in a Graph**
   - **Problem:** Given a graph, find the level (or distance) of each node from a given source node using BFS.
4. **Word Ladder Problem**
   - **Problem:** Given two words (start and end), and a dictionary, find the shortest transformation sequence from start to end such that each transformed word must exist in the dictionary. Each transformation can only change one letter at a time.
5. **Number of Islands**
   - **Problem:** Given a 2D grid representing a map where `'1'` represents land and `'0'` represents water, find the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically.

Depth-First Search (DFS) Problems:

6. **Cycle Detection in an Undirected Graph**
   - **Problem:** Given an undirected graph, detect if there is a cycle using DFS. If a cycle is detected, return `True`; otherwise, return `False`.
7. **Topological Sort of a Directed Acyclic Graph (DAG)**
   - **Problem:** Given a directed acyclic graph (DAG), return a topological ordering of the vertices.
8. **Path Between Two Nodes in a Graph**
   - **Problem:** Given a directed or undirected graph and two nodes, determine if there is a path from the source node to the destination node using DFS.
9. **Count All Paths Between Two Nodes**

    - **Problem:** Given a directed graph and two nodes, count the number of distinct paths between them using DFS.

10. **Detect Cycle in a Directed Graph**
    - **Problem:** Given a directed graph, detect if there is a cycle using DFS. Mark nodes as visited and backtrack to check for cycles.

Topological Sort Problems:

11. **Course Schedule (Topological Sort)**
    - **Problem:** You are given a list of courses and their prerequisites. Each course is a node, and each prerequisite is a directed edge. Determine if it's possible to finish all courses (i.e., check if a valid topological order exists).
12. **Alien Dictionary (Topological Sort)**
    - **Problem:** Given a list of words in an alien dictionary, find the order of characters in the alien language. Assume that the words are sorted lexicographically according to the unknown order.

Bellman-Ford Algorithm Problems:

13. **Single Source Shortest Path with Negative Weights (Bellman-Ford)**
    - **Problem:** Given a graph with possible negative edge weights and a source node, find the shortest path from the source to all other nodes using the Bellman-Ford algorithm.
14. **Detect Negative Weight Cycles (Bellman-Ford)**
    - **Problem:** Given a graph with edge weights, use the Bellman-Ford algorithm to detect if there is a negative weight cycle in the graph.
15. **Find the Longest Path in a Directed Acyclic Graph (DAG)**
    - **Problem:** Given a weighted directed acyclic graph (DAG), find the longest path starting from a source node using dynamic programming combined with topological sorting.