Set 2: Advanced Python and Algorithms

Topic 1: Recursion

Write a recursive function to compute the factorial of a number.
Write a recursive function to compute the nth Fibonacci number.
Write a recursive function to reverse a string.
Write a recursive function to sum the digits of a number.
Write a recursive function to solve the Tower of Hanoi problem for n disks.

Topic 2: Binary Search

Write a function to perform binary search on a sorted list and return the index of the target element (-1 if not found).
Write a function to find the first occurrence of a target element in a sorted list using binary search.
Write a function to find the last occurrence of a target element in a sorted list using binary search.
Write a function to find the number of occurrences of a target element in a sorted list using binary search.
Write a function to find the square root of a number using binary search (integer part only).

Topic 3: Dynamic Programming

Write a function to compute the nth Fibonacci number using dynamic programming.
Write a function to find the length of the longest increasing subsequence in an array.
Write a function to solve the coin change problem (minimum number of coins to make a given amount).
Write a function to solve the 0/1 knapsack problem.
Write a function to find the minimum edit distance between two strings.

Topic 4: Graph Representation

Implement a graph using an adjacency list and write a function to add an edge between two vertices.
Implement a graph using an adjacency matrix and write a function to add an edge between two vertices.
Write a function to convert a graph from adjacency list representation to adjacency matrix representation.
Write a function to convert a graph from adjacency matrix representation to adjacency list representation.
Write a function to check if a graph is connected using adjacency list representation.

Topic 5: Depth-First Search (DFS)

Implement DFS for a graph and print the traversal order.
Write a function to detect a cycle in an undirected graph using DFS.
Write a function to detect a cycle in a directed graph using DFS.
Write a function to find all connected components in an undirected graph using DFS.

Write a function to perform topological sorting on a directed acyclic graph using DFS.

Topic 6: Breadth-First Search (BFS)

Implement BFS for a graph and print the traversal order.
Write a function to find the shortest path from a source vertex to all other vertices in an unweighted graph using BFS.
Write a function to check if a graph is bipartite using BFS.
Write a function to find all connected components in an undirected graph using BFS.
Write a function to perform level-order traversal on a binary tree using BFS.

Topic 7: Merge Sort

Implement the merge sort algorithm to sort a list of integers.
Write a function to merge two sorted arrays into one sorted array.
Write a function to count the number of inversions in an array using merge sort.
Write a function to find the median of two sorted arrays using merge sort.
Write a function to sort a linked list using merge sort.

Topic 8: Quick Sort

Implement the quick sort algorithm to sort a list of integers.
Write a function to find the kth smallest element in an array using quick sort.
Write a function to sort an array of 0s, 1s, and 2s using quick sort.
Write a function to find the partition point in quick sort.
Write a function to perform the three-way partitioning in quick sort.

Topic 9: Two Pointers Technique

Given a sorted array, find two numbers that sum up to a specific target.
Write a function to remove duplicates from a sorted array using the two pointers technique.
Write a function to find the maximum area of water contained between two lines (LeetCode 11: Container With Most Water).
Write a function to move all zeroes to the end of an array while maintaining the relative order of the non-zero elements.
Write a function to find the subarray with the maximum sum in an array (Kadane's algorithm).

Topic 10: Sliding Window Technique

Given an array of integers, find the maximum sum of subarray of length k.
Write a function to find the length of the longest substring without repeating characters.
Write a function to find the minimum window substring that contains all characters of another string.
Write a function to find the maximum number of vowels in a substring of length k.
Write a function to find the longest substring with at most two distinct characters.