

FANG DSA Practice Questions

1. Arrays and Strings

- **Two Sum:** Given an array of integers, return indices of the two numbers such that they add up to a specific target.
- **Longest Substring Without Repeating Characters:** Find the length of the longest substring without repeating characters.
- **Container With Most Water:** Given n non-negative integers a_1, a_2, \dots, a_n , where each represents a point at coordinate (i, a_i) , find two lines, which together with the x-axis forms a container, such that the container contains the most water.
- **Product of Array Except Self:** Given an array `nums` of n integers where $n > 1$, return an array `output` such that `output[i]` is equal to the product of all the elements of `nums` except `nums[i]`.

2. Linked Lists

- **Reverse Linked List:** Reverse a singly linked list.
- **Merge Two Sorted Lists:** Merge two sorted linked lists and return it as a new sorted list.
- **Detect Cycle in a Linked List:** Given a linked list, determine if it has a cycle in it.
- **Palindrome Linked List:** Given a singly linked list, determine if it is a palindrome.

3. Trees and Graphs

- **Lowest Common Ancestor of a Binary Tree:** Given a binary tree, find the lowest common ancestor (LCA) of two given nodes.
- **Binary Tree Maximum Path Sum:** Given a non-empty binary tree, find the maximum path sum.
- **Number of Islands:** Given a 2D grid map of '1's (land) and '0's (water), count the number of islands. An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically.
- **Clone Graph:** Given a reference of a node in a connected undirected graph, return a deep copy (clone) of the graph.

4. Dynamic Programming

- **Longest Increasing Subsequence:** Find the length of the longest increasing subsequence in a given array.

- **Coin Change:** Given an integer array `coins` representing coins of different denominations and an integer `amount`, return the fewest number of coins that you need to make up that amount.
- **Word Break:** Given a string `s` and a dictionary of strings `wordDict`, return `true` if `s` can be segmented into a space-separated sequence of one or more dictionary words.
- **Edit Distance:** Given two strings `word1` and `word2`, return the minimum number of operations required to convert `word1` to `word2`.

5. Sorting and Searching

- **Search in Rotated Sorted Array:** Suppose an array sorted in ascending order is rotated at some pivot unknown to you beforehand. Find the index of a given element.
- **Merge Intervals:** Given a collection of intervals, merge all overlapping intervals.
- **Top K Frequent Elements:** Given a non-empty array of integers, return the `k` most frequent elements.
- **Median of Two Sorted Arrays:** Given two sorted arrays `nums1` and `nums2` of size `m` and `n` respectively, return the median of the two sorted arrays.

6. Backtracking

- **Subsets:** Given a set of distinct integers, return all possible subsets.
- **Combination Sum:** Given a set of candidate numbers (candidates) and a target number (target), find all unique combinations in candidates where the candidate numbers sum to target.
- **N-Queens:** The n-queens puzzle is the problem of placing `n` queens on an `n`×`n` chessboard such that no two queens attack each other.
- **Permutations:** Given an array `nums` of distinct integers, return all the possible permutations.

7. Greedy Algorithms

- **Jump Game:** Given an array of non-negative integers `nums`, you are initially positioned at the first index of the array. Each element in the array represents your maximum jump length at that position. Determine if you can reach the last index.
- **Gas Station:** There are `n` gas stations along a circular route, where the amount of gas at station `i` is `gas[i]`. You have a car with an unlimited gas tank and it costs `cost[i]` of gas to travel from station `i` to its next station $(i + 1) \% n$. Determine if you can travel around the circuit once in the clockwise direction.
- **Task Scheduler:** Given a characters array `tasks`, representing the tasks a CPU needs to do, where each letter represents a different task. Tasks could be done in any order, and each task is done in one unit of time. For each unit of time, the CPU could complete

a task or be idle. Return the least number of units of times that the CPU will take to finish all the given tasks.

- **Candy:** There are n children standing in a line. Each child is assigned a rating value. You are giving candies to these children subjected to the following requirements: Each child must have at least one candy. Children with a higher rating get more candies than their neighbors. Find the minimum number of candies you need to have to distribute the candies to the children.

8. Advanced Topics

- **Trapping Rain Water:** Given n non-negative integers representing an elevation map where the width of each bar is 1, compute how much water it can trap after raining.
- **LRU Cache:** Design and implement a data structure for a Least Recently Used (LRU) cache.
- **Serialize and Deserialize Binary Tree:** Design an algorithm to serialize and deserialize a binary tree.
- **Palindrome Pairs:** Given a list of unique words, find all pairs of distinct indices (i, j) in the given list, so that the concatenation of the two words is a palindrome.

Tips for Solving FANG-Level DSA Questions

1. **Understand the Problem Statement:** Before jumping into coding, ensure you fully understand the problem, the input/output format, and the constraints.
2. **Optimize:** Think about time and space complexity. Aim for the most efficient solution.
3. **Practice:** Regularly solve problems on platforms like LeetCode, HackerRank, or Codeforces.
4. **Mock Interviews:** Practice with peers or use platforms like Pramp or Interviewing.io for mock interviews.
5. **Explain Your Thought Process:** When practicing, explain your thought process aloud as if you were in an interview.