[Go to Mock Interviews]

CODING Interview Result (hard)

Your Score

48/50

The candidate demonstrates exceptional problem-solving skills and a deep understanding of advanced algorithms and data structures, which is highly commendable for a hard-level coding interview. For most questions, they accurately identify the optimal approach, detail the key steps, and provide correct complexity analysis. Their responses are concise, clear, and professional, reflecting strong technical acumen. The only minor area for improvement was in elaborating on the complexity for Q2, but the core idea was sound.

Question-wise Analysis

Question 1

Q1: You are given N items, where each item i has a value v_i and a weight w_i. You are also given a list of M dependencies, where (j, k) means item j must be picked before item k. You can pick at most K items in total. Find the maximum total value you can get, satisfying all dependencies and the item limit.

Answer Quality: Excellent
Relevance: Relevant
Tone & Clarity: Excellent
📝 **Feedback:**

The candidate correctly identifies Bitmask DP as the appropriate technique for N <= 18. The approach of precomputing dependency masks, iterating through all subsets, and validating dependencies and item count is precisely how this problem should be solved. The complexity analysis O(N * 2^N) is accurate and confirms feasibility.

💡 **How to Improve:**

No significant improvement needed for this answer.

Question 2

Q2: You are given N cities and M roads. Each road (u, v, w) connects city u and city v with a travel time w. Some cities have 'fuel stations'. You start at city S with F units of fuel. Each unit of fuel allows you to travel 1 unit of time (costing 1 fuel unit). You can refuel at any fuel station city i. Refueling at city i takes C[i] units of time, and you can refuel up to your tank capacity T (meaning your fuel becomes T, regardless of how much you had). What is the minimum total time (travel + refueling) to reach city D?

Answer Quality: Good
Relevance: Relevant
Tone & Clarity: Good
📝 **Feedback:**

The candidate correctly identifies this as a Dijkstra problem on a state space of (city, fuel_left). The transitions for travel and refueling are correctly defined. However, the complexity analysis of O(N * T) is problematic given T can be 10^9. While the candidate notes 'handled efficiently with pruning', this crucial detail is not elaborated. For such large T, the number of *relevant* fuel levels is usually much smaller (e.g., related to N or M, or only specific values like T, T-w, T-2w, etc.) and should be explicitly explained to justify feasibility.

💡 **How to Improve:**

For problems with large resource constraints (like fuel T), elaborate on how the state space is effectively pruned or how the number of relevant states is bounded, beyond just stating 'handled efficiently with pruning'. For instance, explaining that only discrete fuel levels like T or T-weight_of_an_edge are relevant.

Question 3

Q3: Given N rectangles, where each rectangle is defined by its bottom-left (x1, y1) and top-right (x2, y2) coordinates. All coordinates are integers. Find the total area covered by at least one rectangle.

Answer Quality: Excellent
Relevance: Relevant
Tone & Clarity: Excellent
📝 **Feedback:**

The candidate correctly identifies and concisely explains the standard Sweep Line + Segment Tree approach for this classic problem. The breakdown into events, sorting, segment tree usage with coordinate compression, and area accumulation is perfectly described. The O(N log N) complexity is correct.

💡 **How to Improve:**

No significant improvement needed.

Question 4

Q4: Given a string s, find the length of its longest palindromic substring. Implement an algorithm with O(N) time complexity, where N is the length of the string.

Answer Quality: Excellent
Relevance: Relevant
Tone & Clarity: Excellent
📝 **Feedback:**

The candidate immediately identifies Manacher's Algorithm, which is the specific O(N) solution for this problem. The key steps, including string transformation and efficient expansion using mirror properties, are correctly mentioned. The time and space complexity are also accurately stated.

💡 **How to Improve:**

No significant improvement needed.

Question 5

Q5: You are given N items, each with a weight w_i and a value v_i. You also have K identical bags, each with a maximum weight capacity C. Distribute the items into the bags to maximize the total value. Each item can only be put into one bag. Return the maximum total value.

Answer Quality: Excellent
Relevance: Relevant

Tone & Clarity: Excellent

📝 **Feedback:**

The candidate correctly categorizes this as a multi-bin knapsack problem and identifies Bitmask DP as the suitable approach due to small N. The DP state `dp[i][mask]` and the transition logic, involving disjoint submasks and valid weight constraints, are accurately formulated. The O(K * 3^N) complexity is correct and acceptable for the given constraints.

💡 **How to Improve:**

No significant improvement needed.

📋 Overall Improvement Tips

The candidate has a very strong grasp of algorithms. To further excel, especially in hard problems involving large constraints, always be prepared to justify the feasibility of your complexity analysis by detailing how large parameters (like 'T' in Q2) are handled or bounded within the proposed algorithm. This shows a deeper understanding of practical application and optimization.

Back to Mock Interviews | Print Results