What is Dynamic Programming?
It is an optimization technique that solves complex problems by breaking them down into simpler sub-problems and storing their solutions to avoid redundant calculations.

What are the two main properties of a problem that suggest it can be solved with Dynamic Programming?
The two main properties are *overlapping sub-problems* and *optimal substructure*.

Define the term 'overlapping sub-problems'.
It means that the same smaller problems are solved multiple times, which can be optimized by storing results.

What is optimal substructure in Dynamic Programming?
It implies that an optimal solution to a problem can be constructed from optimal solutions to its sub-problems.

What is the 0/1 Knapsack Problem?
It is a decision problem where items with given weights and values are selected to maximize the total value without exceeding a weight limit, with each item is either included (1) or excluded (0).

How does a 0/1 knapsack problem work?

Create a DP Table with rows as items and columns as weight limits.
Fill the Table by deciding, for each item and weight limit, whether including the item gives a higher value than excluding it.
Extract the Solution from the last cell, which shows the maximum value achievable. Trace back to identify the items included.

How does the 0/1 Knapsack Problem differ from the Fractional Knapsack Problem?
In the 0/1 Knapsack Problem, items are indivisible, while in the Fractional Knapsack Problem, fractions of items can be taken.

Explain the greedy method for solving the Knapsack Problem.
The greedy method selects items based on the highest value-to-weight ratio but is more suitable for the Fractional Knapsack Problem, not the 0/1 variant.

What is the first step in solving the 0/1 Knapsack Problem using Dynamic Programming?
The first step is to create a table with (n+1) rows and (W+1) columns, where n is the number of items and W is the knapsack's weight capacity.

Describe the recursive formula used in the 0/1 Knapsack Dynamic Programming approach.
The formula is T(i, j) = max(T(i-1, j), value_i + T(i-1, j - weight_i)), where T(i,j) is the maximum value for the first i items with a weight limit of j.

What does the last cell in the DP table represent in the 0/1 Knapsack Problem?
It represents the maximum possible value that can be obtained without exceeding the knapsack's weight limit.

How do you trace which items to include in the knapsack after filling the DP table?
Starting from the last cell, trace upward. When an entry value differs from the one above it, mark that item as included in the knapsack.

What is the time complexity of solving the 0/1 Knapsack Problem using Dynamic Programming?
The time complexity is O(nW), where n is the number of items and W is the knapsack capacity.

What is the role of memoization in Dynamic Programming?
Memoization is the technique of storing computed solutions to sub-problems to avoid redundant calculations and improve efficiency.

Can Binary Search be solved using Dynamic Programming? Why or why not?
No, because Binary Search does not have overlapping sub-problems, which is necessary for DP.

Give an example of a problem that can be solved with Dynamic Programming.
Problems like Matrix Chain Multiplication, Longest Common Subsequence, and the 0/1 Knapsack Problem can be solved using DP.

In the context of the 0/1 Knapsack Problem, what does the term 'bottom-up approach' mean?
A bottom-up approach involves building the solution by solving smaller sub-problems first and using their solutions to solve larger sub-problems iteratively.

What is the importance of filling the first row and column with zeros in the DP table for the 0/1 Knapsack Problem?
Filling the first row and column with zeros initializes the base case, where either the weight limit or the number of items is zero.

Explain the significance of the weight constraint in the Knapsack Problem.
The weight constraint ensures that the selected items' combined weight does not exceed the knapsack's capacity, which is a key limitation in the problem.

What is the output of a Dynamic Programming solution for the 0/1 Knapsack Problem?
The output is the maximum value achievable without exceeding the knapsack's weight limit, along with the set of items selected.

Why is the 0/1 Knapsack Problem considered NP-complete?
It is NP-complete because there is no known polynomial-time solution, and it requires evaluating multiple combinations of items, which grows exponentially with input size.