

## **What is a Greedy Algorithm?**

A greedy algorithm solves problems by choosing the best option at each step without considering future consequences. It selects locally optimal choices hoping to find a global optimum.

## **What is the primary characteristic of the Greedy algorithm?**

The algorithm never reverses its earlier decisions, even if they lead to a suboptimal solution. It works in a top-down approach.

## **What is the main advantage of using the Greedy algorithm?**

Greedy algorithms are easier to implement and can often be faster than other algorithms, although they may not always provide the optimal solution.

## **Describe a major drawback of the Greedy algorithm.**

Greedy algorithms may not always provide the optimal solution, as they only make local, immediate choices rather than looking at the overall best solution.

## **What is the Fractional Knapsack problem?**

It involves filling a knapsack with items that have individual weights and values to maximize profit. Unlike the 0/1 knapsack problem, it allows the inclusion of fractions of items.

## **How does the Greedy approach solve the Fractional Knapsack problem?**

It first calculates the value-to-weight ratio for each item, then sorts the items by this ratio in descending order. It then fills the knapsack with items with the highest ratio until the knapsack is full. Add items to the knapsack until full or add a fraction if only part of an item fits.

## **Explain the difference between the Fractional Knapsack and 0/1 Knapsack problems.**

In the Fractional Knapsack problem, items can be broken down and included fractionally. In the 0/1 Knapsack problem, each item can either be included entirely or not at all.

## **Why is the Greedy algorithm suitable for the Fractional Knapsack problem?**

The Fractional Knapsack problem allows fractional inclusion of items, making it suitable for a Greedy approach where items are chosen based on maximum immediate benefit (value-to-weight ratio).

## **In the Fractional Knapsack problem, how is the value-to-weight ratio calculated?**

The value-to-weight ratio is calculated by dividing the value of each item by its weight.

## **How does sorting items by the value-to-weight ratio help in solving the Fractional Knapsack problem?**

Sorting ensures that items offering the highest immediate value are selected first, maximizing the overall profit.

## **If a knapsack has remaining capacity but the next item's weight exceeds it, what does the Greedy algorithm do?**

In the Fractional Knapsack problem, the algorithm takes only a fraction of the item that fits into the remaining capacity of the knapsack.

## **Provide an example of a Greedy method other than the Fractional Knapsack problem.**

One example is the activity selection problem, where the goal is to select the maximum number of non-overlapping activities that can be performed by a single person.

## **What is the time complexity of the Fractional Knapsack problem using the Greedy algorithm?**

The time complexity is  $O(n \log n)$ , where  $n$  is the number of items, due to the need to sort the items based on their value-to-weight ratios.

## **Why is the 0/1 Knapsack problem not solvable with a Greedy algorithm?**

The 0/1 Knapsack problem does not allow fractional items; thus, the Greedy approach may lead to a suboptimal solution as it cannot partially include items.

## **How is the optimal solution calculated once the items are sorted by ratio in the Fractional Knapsack problem?**

Items are added to the knapsack in order of their sorted ratios, and the total profit is calculated by summing the full or fractional values of the items added.

## **What happens if all items have the same value-to-weight ratio in the Fractional Knapsack problem?**

If all items have the same value-to-weight ratio, the Greedy algorithm can select items in any order, as all would contribute equally to the total value.

## **What kind of problems are best suited for Greedy algorithms?**

Problems where locally optimal choices lead to a globally optimal solution, such as scheduling problems, shortest path in a weighted graph (Dijkstra's algorithm), and the Fractional Knapsack problem.

**What is the result when using a Greedy algorithm for a problem that does not satisfy the Greedy-choice property?**

The algorithm may produce a suboptimal solution, as it will only provide the best immediate choice rather than the best overall solution.