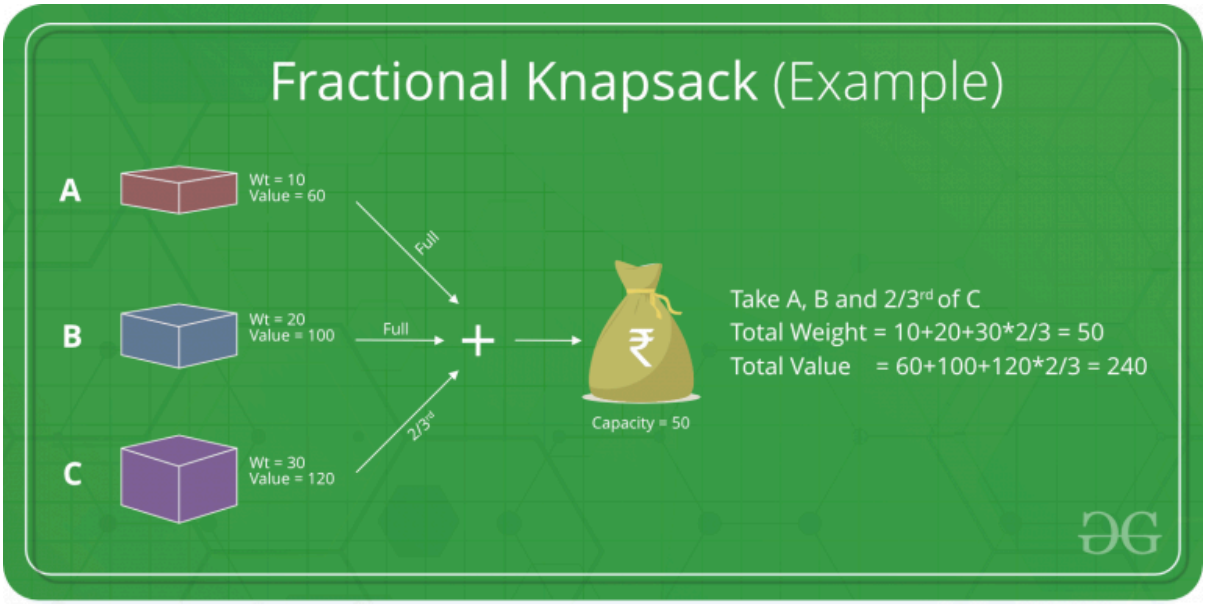
Greedy Algorithms

Last Updated : 30 Jul, 2024

Greedy algorithms are a class of algorithms that make **locally optimal** choices at each step with the hope of finding a **global optimum** solution. In these algorithms, decisions are made based on the information available at the current moment without considering the consequences of these decisions in the future. The key idea is to select the best possible choice at each step, leading to a solution that may not always be the most optimal but is often good enough for many problems.

In this article, we will understand greedy algorithms with examples. We will also look at problems and their solutions using the greedy approach.



Greedy Algorithms

Table of Content

- Disadvantages/Limitations of Using a Greedy Algorithm
- Basics of Greedy Algorithm
- Standard Greedy Algorithms
- Greedy Problems on Array
- Greedy Problems on Operating System
- Greedy Problems on Graph
- Approximate Greedy Algorithm for NP Complete
- Greedy for Special cases of DP
- Easy Problems on Greedy Algorithm
- Medium Problems on Greedy Algorithm
- Hard Problems on Greedy Algorithm

What is Greedy Algorithm?

A **greedy algorithm** is a type of optimization algorithm that makes locally optimal choices at each step to find a globally optimal solution. It operates on the principle of "taking the best option now" without considering the long-term consequences.

To learn what is greedy method and how to use the greedy approach, read the given tutorial on the Greedy Algorithm:

Greedy Algorithm Tutorial

Steps for Creating a Greedy Algorithm

The steps to define a greedy algorithm are:

1. **Define the problem:** Clearly state the problem to be solved and the objective to be optimized.
2. **Identify the greedy choice:** Determine the locally optimal choice at each step based on the current state.
3. **Make the greedy choice:** Select the greedy choice and update the current state.
4. **Repeat:** Continue making greedy choices until a solution is reached.

Following the given steps, one can learn how to use greedy algorithms to find optimal solutions.

Greedy Algorithm Examples

Examples of greedy algorithms are the best way to understand the algorithm. Some greedy algorithm real-life examples are:

- **Fractional Knapsack:** Optimizes the value of items that can be fractionally included in a knapsack with limited capacity.
- **Dijkstra's algorithm:** Finds the shortest path from a source vertex to all other vertices in a weighted graph.
- **Kruskal's algorithm:** Finds the minimum spanning tree of a weighted graph.
- **Huffman coding:** Compresses data by assigning shorter codes to more frequent symbols.

Applications of Greedy Algorithm

There are many **applications of the greedy method in DAA**. Some important greedy algorithm applications are:

- Assigning tasks to resources to minimize waiting time or maximize efficiency.
- Selecting the most valuable items to fit into a knapsack with limited capacity.
- Dividing an image into regions with similar characteristics.
- Reducing the size of data by removing redundant information.

Disadvantages/Limitations of Using a Greedy Algorithm

Below are some disadvantages of the Greedy Algorithm:

- Greedy algorithms may not always find the best possible solution.
- The order in which the elements are considered can significantly impact the outcome.

- Greedy algorithms focus on local optimizations and may miss better solutions that require considering a broader context.
- Greedy algorithms are not applicable to problems where the greedy choice does not lead to an optimal solution.

Basics of Greedy Algorithm:

- Introduction to Greedy Algorithm – Data Structures and Algorithm Tutorials
- Greedy Algorithms (General Structure and Applications)
- Difference between Greedy Algorithm and Divide and Conquer Algorithm
- Greedy approach vs Dynamic programming
- Comparison among Greedy, Divide and Conquer and Dynamic Programming algorithm

Standard Greedy Algorithms:

- Activity Selection Problem
- Job Sequencing Problem
- Huffman Coding
- Huffman Decoding
- Water Connection Problem
- Minimum Swaps for Bracket Balancing
- Egyptian Fraction
- Policemen catch thieves
- Fitting Shelves Problem
- Assign Mice to Holes

Greedy Problems on Array:

- Minimum product subset of an array
- Maximize array sum after K negations using Sorting
- Minimum sum of product of two arrays
- Minimum sum of absolute difference of pairs of two arrays

- Minimum increment/decrement to make array non-Increasing
- Sorting array with reverse around middle
- Sum of Areas of Rectangles possible for an array
- Largest lexicographic array with at-most K consecutive swaps
- Partition into two subarrays of lengths k and (N – k) such that the difference of sums is maximum

Greedy Problems on Operating System:

- First Fit algorithm in Memory Management
- Best Fit algorithm in Memory Management
- Worst Fit algorithm in Memory Management
- Shortest Job First Scheduling
- Job Scheduling with two jobs allowed at a time
- Program for Optimal Page Replacement Algorithm

Greedy Problems on Graph:

- Kruskal's Minimum Spanning Tree
- Prim's Minimum Spanning Tree
- Boruvka's Minimum Spanning Tree
- Dijkastra's Shortest Path Algorithm
- Dial's Algorithm
- Minimum cost to connect all cities
- Max Flow Problem Introduction
- Number of single cycle components in an undirected graph

Approximate Greedy Algorithm for NP Complete:

- Set cover problem
- Bin Packing Problem
- Graph Coloring
- K-centers problem
- Shortest superstring problem
- Approximate solution for Travelling Salesman Problem using MST

Greedy for Special cases of DP:

- Fractional Knapsack Problem
- Minimum number of coins required

Easy Problems on Greedy Algorithm:

- Split n into maximum composite numbers
- Buy Maximum Stocks if i stocks can be bought on i-th day
- Find the minimum and maximum amount to buy all N candies
- Maximum sum possible equal to sum of three stacks
- Divide cuboid into cubes such that sum of volumes is maximum
- Maximum number of customers that can be satisfied with given quantity
- Minimum rotations to unlock a circular lock
- Minimum rooms for m events of n batches with given schedule
- Minimum cost to make array size 1 by removing larger of pairs
- Minimum cost for acquiring all coins with k extra coins allowed with every coin
- Minimum increment by k operations to make all elements equal
- Find minimum number of currency notes and values that sum to given amount
- Smallest subset with sum greater than all other elements

Medium Problems on Greedy Algorithm:

- Maximum trains for which stoppage can be provided
- Minimum Fibonacci terms with sum equal to K
- Divide 1 to n into two groups with minimum sum difference
- Paper cut into minimum number of squares
- Minimum difference between groups of size two
- Connect n ropes with minimum cost
- Minimum number of Platforms required for a railway/bus station
- Minimum initial vertices to traverse whole matrix with given conditions

- Largest palindromic number by permuting digits
- Find Smallest number with given number of digits and digits sum
- Lexicographically largest subsequence such that every character occurs at least k times

Hard Problems on Greedy Algorithm:

- Maximum elements that can be made equal with k updates
- Minimize cash flow among friends
- Minimum Cost to cut a board into squares
- Minimum cost to process m tasks where switching costs
- Minimum time to finish all jobs with given constraints
- Minimize the maximum difference between the heights of towers
- Minimum edges to reverse to make path from a source to a destination
- Find the Largest Cube formed by Deleting minimum Digits from a number
- Rearrange characters in a string such that no two adjacent are same
- Rearrange a string so that all same characters become d distance away

Quick Links:

- Learn Data Structure and Algorithms | DSA Tutorial
- Top 20 Greedy Algorithms Interview Questions
- 'Practice Problems' on Greedy Algorithms
- 'Quiz' on Greedy Algorithms

Previous Article

Backtracking Algorithm

Next Article

Dynamic Programming or DP