Day 21 Greedy

ITSRUNTYM

What is the Greedy Approach in DSA?
The **Greedy Algorithm** is a problem-solving strategy where, **at each step**, you make the choice that **looks best at the moment**, with the hope that this leads to the **global optimum** solution.
Instead of considering all possible solutions (like in Dynamic Programming or Backtracking), greedy algorithms make **locally optimal choices** in a **step-by-step** manner.

Key Characteristics of Greedy Algorithms

Feature	Description
Local Optimization	Greedy chooses the best option at each step.
No Backtracking	Once a decision is made, it's never changed.
Fast Execution	Usually more efficient (O(n log n) or better).
Doesn't Always Work	Only works when a **greedy choice property** and **optimal substructure** exist.

Conditions for Applying Greedy

1. **Greedy Choice Property**
   We can make a local optimal choice, and it leads to a globally optimal solution.
2. **Optimal Substructure**
   A problem has optimal substructure if its solution can be constructed from solutions of its subproblems.

Real-life Example

Activity Selection Problem
You are given n activities with their start and end times. Select the **maximum number** of activities that don't overlap.

Greedy Approach:

- Sort by **end time** of activities.
- Always pick the **next activity** with the earliest finish time that doesn't overlap.

Steps in Designing a Greedy Algorithm:

1. **Understand the problem's structure.**
2. **Define the objective function.**
3. **Make a greedy choice (best current option).**
4. **Reduce the problem.**
5. **Repeat until a solution is built.**

When to Use Greedy:

Use when:

- You want a quick solution.
- You can prove that the greedy choice leads to an optimal solution.
- The problem satisfies the **greedy-choice property** and **optimal substructure**.

Problem Name	Description	Link
Assign Cookies	Assign cookies to children to maximize satisfaction	🔗 LeetCode #455
Non-overlapping Intervals	Remove the minimum number of intervals to eliminate overlaps	🔗 LeetCode #435
Jump Game	Can you reach the last index by making max jumps?	🔗 LeetCode #55
Jump Game II	Minimum number of jumps to reach last index	🔗 LeetCode #45
Gas Station	Find starting station to complete the circuit	🔗 LeetCode #134
Candy	Distribute candies to children with ratings	🔗 LeetCode #135
Merge Triplets to Form Target Triplet	Check if you can form target triplet using rules	🔗 LeetCode #1899
Minimum Number of Arrows to Burst Balloons	Burst all balloons using minimum arrows	🔗 LeetCode #452
Reconstruct Queue by Height	Arrange people based on height and position	🔗 LeetCode #406

Task Scheduler	Minimize total time to finish all tasks with cooldown	🔗 LeetCode #621