Greedy Algorithm Problems for Beginners

(HackWithInfy Round 1 Style)

1. Easy 1: Minimum Coins (Greedy Applicable Denominations Only)

Problem Description:
You are given an amount and a list of coin denominations. Your task is to find the minimum number of coins required to make up the given amount using a greedy approach (assume denominations allow greedy solution to work correctly).

**Find the minimum number of coins required to make a given amount using the provided denominations.
(Greedy works only if denominations follow the Indian currency system or descending ratio-friendly coins.)

Important:
*The **greedy solution does not always give the optimal result** for this problem unless the coin denominations are **canonical** (like Indian currency: {1, 2, 5, 10, 20, 50, 100, 500, 2000}).*

Constraints:

- `1 <= amount <= 10^6`
- `1 <= coin_i <= amount`
- Coins are provided in descending order.

Input Format:

```
amount
coin1 coin2 coin3 ... (space separated)
```

Output Format:

```
Minimum coins needed
```

Sample:

```
Input:
43
10 5 2 1

Output:
```

```
6
```

Explanation:
Coins used: 4×10 + 1×2 + 1×1 = 43 (total coins = 6).

Edge Cases:

- Amount = 0 → Output should be 0
- Amount smaller than smallest coin → Must return -1 if no solution (optional variant)

LeetCode Reference:

- Coin Change 2
- Greedy Version on GFG

YouTube Explanation:

- Greedy Coin Change Explained – by Abdul Bari
- Greedy Algorithm | Coin Change Problem – by GeeksforGeeks

2 .Easy 2: Activity Selection

Problem Description:
Given start and end times of activities, select the maximum number of non-overlapping activities. Activities are selected based on earliest finishing time.

Constraints:

- `1 <= n <= 10^4`
- `0 <= start_i < end_i <= 10^6`

Input Format:

```
n
start1 end1
start2 end2
...
```

Output Format:

```
Maximum number of activities
```

Sample:

```
Input:
6
```

```
1 2
3 4
0 6
5 7
8 9
5 9

Output:
4
```

Explanation:
Selected: (1,2), (3,4), (5,7), (8,9)

Edge Cases:

- Activities completely overlapping → Only one can be selected
- All activities non-overlapping → All can be selected

LeetCode Reference:

- Non-overlapping Intervals
- Maximum Number of Events That Can Be Attended

YouTube Explanation:

- Activity Selection Problem - Greedy Algorithm – by GeeksforGeeks
- Activity Selection | C++ Solution – by take U forward

3.Find the Minimum Number of Fibonacci Numbers Whose Sum Is K

- **LeetCode Problem:** 1414. Find the Minimum Number of Fibonacci Numbers Whose Sum Is K
- **YouTube Solution:** Find Minimum Number of Fibonacci Numbers

Problem Description:

Given an integer `k`, return the minimum number of Fibonacci numbers whose sum is equal to `k`.

Each Fibonacci number can be used **at most once**.

Greedy Approach:
Always pick the **largest Fibonacci number less than or equal to the remaining sum.**

Input Format:

- A single integer `k`

Output Format:

- Minimum number of Fibonacci numbers required to sum up to `k`

Constraints:

- $1 \leq k \leq 10^9$

Sample Input:

```
19
```

Sample Output:

```
3
```

Explanation:

19 = 13 + 5 + 1
Minimum 3 Fibonacci numbers.

💡 Greedy Concept:

At each step, choose the **largest possible Fibonacci number** that does not exceed the remaining sum.
This is similar to a coin change problem where greedy gives the optimal solution because Fibonacci numbers grow fast.

Additional:

- You can **precompute** all Fibonacci numbers less than or equal to `k` (there are at most 44 Fibonacci numbers less than 10^9).
- This is a **must-solve** for greedy understanding as it's a classic example where greedy is optimal.

4. Medium 1: Fractional Knapsack

Problem Description:
Given `n` items with their values and weights, maximize the total value in a knapsack of capacity `W`. You can take fractional parts of items.

Constraints:

- `1 <= n <= 10^4`
- `1 <= W <= 10^6`
- `1 <= value_i, weight_i <= 10^4`

Input Format:

```
W
n
value1 weight1
value2 weight2
...
```

Output Format:

```
Maximum value (rounded to 2 decimal places)
```

Sample:

```
Input:
50
3
60 10
100 20
120 30

Output:
240.00
```

Explanation:
Take full items 1 & 2, and 2/3 of item 3.

Edge Cases:

- W = 0 → Maximum value is 0
- All items too heavy → Only fractions can be taken

5. Medium 2: Train Platforms

Problem Description:
Given arrival and departure times of trains, find the minimum number of platforms required so that no train waits.

Constraints:

- `1 <= n <= 10^4`
- Time format: HHMM (e.g., 900 for 9:00 AM)

Input Format:

```
n
arrival1 departure1
arrival2 departure2
...
```

Output Format:

```
Minimum number of platforms required
```

Sample:

```
Input:
6
900 910
940 1200
950 1120
1100 1130
1500 1900
1800 2000

Output:
3
```

Explanation:
Maximum overlap occurs when 3 trains are present simultaneously.

Edge Cases:

- All trains arriving and leaving at the same time → Need n platforms
- Completely non-overlapping → Only one platform needed

6. Hard 1: Job Sequencing with Deadlines

Problem Description:
You are given `n` jobs with deadlines and profits. Each job takes exactly 1 unit of time. Schedule jobs to maximize total profit. Jobs must be completed before their deadline.

Constraints:

- `1 <= n <= 10^4`
- `1 <= deadline_i <= 1000`
- `1 <= profit_i <= 10^4`

Input Format:

```
n
```

```
deadline1 profit1
deadline2 profit2
...
```

Output Format:

```
Number of jobs scheduled
Total profit
```

Sample:

```
Input:
4
4 20
1 10
1 40
1 30

Output:
2
60
```

Explanation:
Select jobs with highest profit that can be completed on time.

Edge Cases:

- Deadlines all 1 → Only one job can be selected
- Deadlines sufficiently large → Can schedule all jobs

7 .Hard 2: Car Refueling Problem

Problem Description:
You need to travel to a target distance starting with initial fuel. Along the way, there are gas stations at certain distances. Find the minimum number of refuels required to reach the destination. If not possible, return -1.

Constraints:

- `1 <= target <= 10^9`
- `0 <= start_fuel <= target`
- `0 <= station_i <= target`
- `1 <= station_fuel_i <= 10^4`

Input Format:

```
target
start_fuel
n
position1 fuel1
position2 fuel2
...
```

Output Format:

```
Minimum number of refuels (or -1 if impossible)
```

Sample:

```
Input:
100
10
4
10 60
20 30
30 30
60 40

Output:
2
```

Explanation:
Optimal stops: Station at 10 (add 60 fuel), station at 60 (add 40 fuel).

Edge Cases:

- No stations → If start fuel is enough, return 0; else -1
- Stations at same location → Handle duplicates

Here's your **complete problem set with both LeetCode and YouTube video explanations added for each problem:**

🚀 Greedy Algorithm Problems for Beginners with LeetCode & YouTube Links

(HackWithInfy Round 1 Style)

✅ Easy 1: Minimum Coins

Problem Description:
Given an amount and coin denominations, find the minimum number of coins required using a greedy approach.

✅ Easy 2: Activity Selection

Problem Description:
Select the maximum number of non-overlapping activities based on start and end times.

LeetCode Reference:

- Non-overlapping Intervals
- Maximum Number of Events That Can Be Attended

YouTube Explanation:

- Activity Selection Problem - Greedy Algorithm – by GeeksforGeeks
- Activity Selection | C++ Solution – by take U forward

✅ Medium 1: Fractional Knapsack

Problem Description:
Maximize total value by taking fractional parts of items.

LeetCode Reference:

- Fractional Knapsack on GFG

YouTube Explanation:

- Fractional Knapsack | Greedy Algorithm – by take U forward
- Fractional Knapsack Problem | Greedy Algorithm – by GeeksforGeeks

---

✅ Medium 2: Train Platforms

Problem Description:
Find the minimum number of platforms required so no train waits.

LeetCode Reference:

- Meeting Rooms II
- Train Platform Problem on GFG

YouTube Explanation:

- Minimum Number of Platforms Needed – by take U forward

- Train Platform Problem | Greedy – by GeeksforGeeks

✅ Hard 1: Job Sequencing with Deadlines

Problem Description:
Schedule jobs to maximize profit within their deadlines.

LeetCode Reference:

- Maximum Profit in Job Scheduling
- Job Sequencing Problem on GFG

YouTube Explanation:

- Job Sequencing Problem Explained – by take U forward
- Job Sequencing Problem | Greedy – by GeeksforGeeks

✅ Hard 2: Car Refueling Problem

Problem Description:
Minimum refueling stops needed to reach the target distance.

LeetCode Reference:

- Minimum Number of Refueling Stops

YouTube Explanation:

- Car Refueling Problem Explained – by NeetCode
- Minimum Number of Refueling Stops (Heap Approach) – by Coding Decode

---

📂 Summary Table with Resources:

Problem	LeetCode Link	YouTube Link
Minimum Coins	Link	Abdul Bari / GFG
Activity Selection	Link	GFG / take U forward
Fractional Knapsack	Link	take U forward / GFG
Train Platforms	Link	take U forward / GFG
Job Sequencing	Link	take U forward / GFG
Car Refueling	Link	NeetCode / Coding Decode

If you want, I can also:

- Prepare **starter Python templates** for each question.
- Design a **Google Sheet tracker** for progress monitoring.
- Suggest **mock test schedules** for each week.

Let me know if you'd like to proceed!