Pick-Not Pick (Include-Exclude) Problem Sheet

Pick-Not Pick (Include-Exclude) paradigm. Each problem includes a detailed explanation, input/output format, constraints, example with explanation, and edge cases.

---

1. Weighted Interval Scheduling

Problem: Maximum Profit in Job Scheduling

Description: Given jobs with start time, end time, and profit, schedule jobs to maximize profit such that no two jobs overlap.

Input Format:

- startTime[]: array of integers
- endTime[]: array of integers
- profit[]: array of integers

Output Format:

- Maximum profit achievable

Constraints:

- 1 <= startTime.length <= 5 * 10^4
- 1 <= profit[i] <= 10^4

Example: Input: startTime = [1,2,3,3] endTime = [3,4,5,6] profit = [50,10,40,70]

Output: 120

Explanation: Pick jobs 1 and 4 (profit = 50 + 70).

Practice Link: LeetCode 1235

---

2. Maximum Length of Pair Chain

Description: Find the longest chain you can form with pairs where for (a, b) and (c, d), b < c.

Input Format:

- List of pairs: [[a1, b1], [a2, b2], ...]

Output Format:

- Maximum chain length

Constraints:

- 1 <= pairs.length <= 1000

Example: Input: [[1,2],[2,3],[3,4]]

Output: 2

Explanation: Longest chain is [1,2] -> [3,4].

Practice Link: LeetCode 646

---

3. Counting 0/1 Subset Sum (Target Sum)

Description: Given an array, count the number of ways to assign '+' or '-' to reach a target sum.

Input Format:

- nums[]: array of integers
- target: integer

Output Format:

- Number of ways

Constraints:

- 1 <= nums.length <= 20

Example: Input: nums = [1,1,1,1,1], target = 3

Output: 5

Practice Link: LeetCode 494

---

4. Longest Common Subsequence (LCS)

Description: Find the length of the longest common subsequence between two strings.

Input Format:

- text1: string
- text2: string

Output Format:

- Integer (LCS length)

Constraints:

- 1 <= text1.length, text2.length <= 1000

Example: Input: text1 = "abcde", text2 = "ace"

Output: 3

Practice Link: LeetCode 1143

---

5. Minimum Deletions for Divisibility by K

Description: Minimize deletions to make the sum divisible by 3.

Input Format:

- digits[]: array of integers (0-9)

Output Format:

- Minimum deletions (if possible)

Constraints:

- 1 <= digits.length <= 1000

Example: Input: digits = [8,1,9]

Output: 0 (Sum = 18 is divisible by 3)

Practice Link: LeetCode 1363

---

6. Delete and Earn

Description: Pick a number (add total x * freq[x], skip x-1/x+1) or not pick to maximize sum.

Input Format:

- nums[]: array of integers

Output Format:

- Maximum points earned

Constraints:

- 1 <= nums.length <= 2 * 10^4

Example: Input: nums = [3,4,2]

Output: 6 (Pick 4, skip 3, pick 2)

Practice Link: LeetCode 740

---

7. House Robber II (Circular)

Description: House Robber problem with circular arrangement.

Input Format:

- nums[]: array of integers

Output Format:

- Maximum sum possible

Constraints:

- 1 <= nums.length <= 100

Example: Input: nums = [2,3,2]

Output: 3

Practice Link: LeetCode 213

---

8. Counting Binary Strings without Consecutive 1s

Description: Count the number of binary strings of length n without consecutive 1s.

Input Format:

- Integer n

Output Format:

- Number of valid strings

Constraints:

- 1 <= n <= 10^9

Example: Input: n = 3

Output: 5

Practice Link: LeetCode 600

---

9. Painting Fence (k Colors)

Description: Count ways to paint n fences with k colors so that no more than two adjacent fences have the same color.

Input Format:

- Integers n (fences) and k (colors)

Output Format:

- Number of ways

Constraints:

- 1 <= n <= 50, 1 <= k <= 100

Example: Input: n = 3, k = 2

Output: 6

Practice Link: LeetCode 276

10. Minimum Refueling Stops

Description: Given target distance and stations, find the minimum number of refueling stops.

Input Format:

- target: integer
- startFuel: integer
- stations[][]: array of [position, fuel]

Output Format:

- Minimum number of stops (or -1 if impossible)

Constraints:

- 1 <= stations.length <= 500

Example: Input: target = 100, startFuel = 10, stations = [[10,60],[20,30],[30,30],[60,40]]

Output: 2

Practice Link: LeetCode 871