

Here are **30 one-line Codeforces-style problem statements** that involve **Binary Search** and commonly follow the "**minimum/maximum satisfying condition**" pattern:

1. Find the minimum value of x such that you can complete all tasks within d days with speed x .
2. Given a budget, find the maximum number of items you can buy from a sorted price list.
3. Find the minimum maximum distance you need to jump across stones to reach the end.
4. Find the minimum capacity required to ship all packages within D days.
5. Determine the minimum time needed to print n books with k printers of different speeds.
6. Find the smallest x such that the sum of floors of a_i / x is less than or equal to m .
7. Given n ropes, find the maximum length L such that you can cut at least k ropes of length L .
8. Find the smallest possible radius of light so that all houses are covered with given number of lanterns.
9. Find the largest minimum distance to place k cows in n stalls.
10. Given a square of side s , find the maximum number of tiles of size $a \times a$ that fit.
11. Given array and limit m , find minimum largest subarray sum after splitting into k parts.
12. Given budget and price function, find the maximum number of units you can buy.
13. Given n planks of wood, find the maximum height h such that after cutting, the total wood is at least m .
14. Find the minimum number of steps to reach at least k coins using up to n moves and doubling strategy.
15. Find the smallest x such that at least k numbers in array are $\leq x$.
16. Given a sorted list and a target, find the minimum index i such that $a[i] \geq \text{target}$.
17. Find the maximum length L of a contiguous subarray such that $\text{sum} \leq s$.
18. Find the maximum number of contests you can attend without overlapping using binary search over time.
19. Given n boxes and max weight limit, find the minimum number of trips needed.
20. Determine the minimum fuel tank capacity required to reach the destination with given fuel stations.
21. Find the minimum number of steps to reduce n to 1 using allowed operations with a cost.
22. Find the minimum height at which the total harvested grain is $\geq k$.
23. Find the earliest day where you can collect at least k flowers from a garden.
24. Given n tasks and k processors, find the minimal time to complete all tasks.
25. Determine the maximum speed v at which a train can go without violating speed limits.
26. Find the minimum value x such that a graph can be built with total weight $\leq x$.
27. Find the minimum maximum load across all workers when distributing jobs.
28. Determine the minimum power level required to defeat all enemies with special rules.
29. Find the smallest number of moves needed to reach a target sum using available steps.
30. Find the earliest possible time to cross a river with changing tide heights.

Each of these is inspired by real Codeforces or competitive programming problems where **binary search on the answer** is the key technique. Want full problems or solutions for any of them?

Ask ChatGPT

You said:

Two Pointers For range/subarray problems (e.g., max length with some condition).

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** involving the **Two Pointers** technique, especially useful in **range/subarray problems** (e.g., max/min length, counting ranges with some condition):

✓ Two Pointers: 30 One-Liners for Codeforces Practice

1. Find the length of the longest subarray with all distinct elements.
2. Count the number of subarrays with sum $\leq k$.
3. Find the maximum length of a subarray such that the difference between max and min $\leq k$.
4. Count the number of pairs (i, j) such that $i < j$ and $a[i] + a[j] \leq k$.
5. Count the number of subarrays with at most k distinct integers.
6. Find the length of the longest subarray with sum exactly equal to k .
7. Find the minimum window in $a[]$ that contains all elements of $b[]$.
8. Count the number of subarrays where all elements are less than or equal to k .
9. Find the number of contiguous subarrays where product of elements $< k$.
10. Given two sorted arrays, count number of pairs with sum $\leq x$.
11. Count the number of good segments where the sum is divisible by k .
12. Count the number of subarrays with alternating even-odd or odd-even elements.
13. Find the maximum length of subarray with equal number of 0s and 1s.
14. Given array and integer k , count subarrays with at most k inversions.
15. Count the number of subarrays where the sum is in range $[L, R]$.
16. Find the longest subarray that can be made non-decreasing by removing at most one element.
17. Find the number of triplets (i, j, k) such that $i < j < k$ and values form arithmetic progression.
18. Find the longest subarray where the number of 1s is at most k .
19. Given n days of prices, find the number of subarrays where the price never decreases.
20. Count the number of subarrays of size k where the average is greater than x .
21. Find maximum length of subarray with sum divisible by k .
22. Find the number of index pairs (i, j) such that $a[i] * b[j] \leq x$ using sorted arrays.
23. Find the number of subarrays with exactly k distinct elements.
24. Count number of substrings where the number of vowels and consonants are equal.

25. Given array $a[]$, find the longest segment where all values appear at most k times.
26. Count the number of subarrays of size k with all unique elements.
27. Find the longest prefix which is also a suffix using two pointers.
28. Count the number of non-overlapping intervals in array where $\text{sum} \leq k$.
29. Count all unique subarrays with sorted order.
30. Find the number of subarrays where the absolute difference between adjacent elements is $\leq d$.

These problems are ideal for learning **sliding window techniques** using two pointers. Want explanations, difficulty levels, or full problems? Just ask!

You said:

Prefix Sum / Suffix Sum / Difference Array Used in range-based counting and optimizations

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** based on:

✓ **Prefix Sum / Suffix Sum / Difference Array** techniques — used in **range-based counting, optimizations, or updates**.

🌀 Prefix Sum / Suffix Sum / Difference Array (30 Problem Ideas)

1. Given an array, answer q queries to find the sum of elements from l to r .
2. Count the number of subarrays whose sum is divisible by k .
3. Given a binary string, find the number of substrings with equal number of 0s and 1s.
4. Find the maximum subarray sum using prefix sums.
5. Given an array and q range increment operations, compute the final array using a difference array.
6. For a given array, find the number of pairs (i, j) such that sum of elements between them is even.
7. Given an array of heights, find the minimum total operations to make all elements equal using prefix sums.
8. Given a string, find the number of palindromic substrings using prefix count of characters.
9. Answer q queries: count number of elements $\leq k$ in range $[l, r]$.
10. Given n elements and q range increment operations, find the index with the maximum value.
11. Count the number of subarrays where sum is $\leq k$ using prefix sums + binary search.
12. Find the minimum number of changes needed to make the sum of any subarray of length k even.
13. Given a matrix, answer queries to find the sum of submatrices using 2D prefix sums.
14. For each i , find the number of elements less than $a[i]$ to its left using prefix counts.
15. Given an array, find the smallest subarray with sum $\geq s$ using prefix sums.

16. Construct an array from q range addition queries using difference array and output max element.
17. Answer multiple queries to count how many times each index is updated in a set of range updates.
18. Count the number of subarrays whose XOR is zero using prefix XOR.
19. Given daily temperatures, find for each day how many days until a warmer temperature using suffix.
20. Answer q queries: compute the product of elements in range $[l, r]$ modulo 10^9+7 using prefix products.
21. Given two arrays, answer how many times sum of prefix of a equals prefix of b .
22. Given an array, find number of subarrays where the number of 1s $>$ number of 0s.
23. Preprocess prefix sum to quickly answer min/max value in a subarray.
24. Find the longest subarray with equal number of positive and negative numbers.
25. Using prefix sum, answer queries to check if sum of $[l, r]$ is a multiple of x .
26. Using a difference array, perform multiple range additions and then output the entire array.
27. Count number of ways to partition array into three parts with equal sum using prefix sums.
28. Find the maximum length of a prefix that can be rearranged into a palindrome.
29. Given q range flip operations on a binary array, apply all in $O(n + q)$ using difference array.
30. In a game, apply health changes over ranges of enemies using a difference array efficiently.

Want the actual Codeforces problem links for these patterns or explanations with solutions? Just let me know!

You said:

Greedy + Sorting Sorting and choosing based on conditions is a classic pattern.

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** based on the **Greedy + Sorting** pattern — one of the most **classic and powerful strategies** in competitive programming.

✂ Greedy + Sorting – 30 Problem Ideas

1. You are given n tasks with deadlines and durations — complete the maximum number before their deadlines.
2. Given meeting start and end times, find the maximum number of non-overlapping meetings.
3. You have n items with weights and values — maximize value in a bag of limited capacity (Fractional Knapsack).
4. Assign n jobs to m machines such that the maximum load on any machine is minimized.

5. Given n children and m candies, assign minimum candies so each child gets \geq their requirement.
6. Find the minimum number of platforms needed at a station given train arrival/departure times.
7. Choose the minimum number of intervals to cover a given range $[0, T]$.
8. Find the minimum number of coins/notes to make sum x from given denominations.
9. You have n lectures with start and end times, find the maximum number you can attend.
10. Buy items with given prices to maximize the number of items you can purchase with budget k .
11. Assign n tasks to n workers such that total time is minimized (sort both lists and pair optimally).
12. Given intervals, remove the minimum number so that the rest are non-overlapping.
13. You are given n ropes — connect all ropes into one with minimal total cost.
14. Schedule jobs such that the total penalty for missing deadlines is minimized.
15. Given list of people with weight and strength, stack them to form the highest possible tower.
16. Choose projects with start and end times to maximize total reward.
17. Find the lexicographically smallest string by removing characters with a cost limit.
18. Given a list of events with profit and deadlines, pick maximum profit subset.
19. Maximize the number of people who can cross the bridge without exceeding total weight.
20. Given n items with $(\text{value}, \text{time})$ pairs, select maximum total value under total time T .
21. Find the minimum number of swaps to sort an array using allowed operations.
22. Arrange people in a queue to minimize the sum of their waiting times.
23. Choose k pairs $(a[i], b[i])$ such that total $a[i]/b[i]$ ratio is minimized.
24. Divide array into k subsets of equal sum — sort and use greedy assignment.
25. Find the minimum cost to hire k workers where each worker has a wage and quality.
26. Maximize total profit by choosing jobs and assigning limited number of workers.
27. You can bribe your way forward in a queue — calculate the minimum number of bribes or print "Too chaotic".
28. You can move right or down in a grid with costs — find the cheapest path using greedy + sorting.
29. Match n applicants with m apartments where the apartment size can differ by at most k .
30. Given soldiers with strength and weapons with power, assign to maximize total win count.

These problems require sorting the data based on some key, then **greedily selecting** the best option at each step (e.g., earliest deadline, smallest cost, highest value/weight ratio, etc.).

Would you like explanations or real Codeforces links for any of these?

You said:

DFS/BFS on small graphs or grids Needed for basic graph traversal questions.

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** based on **DFS/BFS on small graphs or grids**, commonly used for **basic graph traversal** tasks:

🔍 DFS/BFS on Small Graphs or Grids – 30 Problem Ideas

1. Count the number of connected components in an undirected graph.
2. Given a grid with obstacles, find the shortest path from top-left to bottom-right.
3. Check if an undirected graph is a tree.
4. Find the number of islands in a $n \times m$ binary grid ($1 = \text{land}$, $0 = \text{water}$).
5. Determine if there is a cycle in an undirected graph.
6. Given a maze, find if there's a path from the entrance to the exit.
7. Color a graph with 2 colors to check if it's bipartite.
8. In a grid of letters, find whether a given word exists following adjacent cells.
9. Count the number of enclosed regions (surrounded regions) in a 2D grid.
10. From a source node, find the minimum number of steps to reach each node in an unweighted graph.
11. Find the longest path in a tree.
12. For each node, find the size of the connected component it belongs to.
13. Count how many distinct paths exist between two vertices in a directed acyclic graph (DAG).
14. Detect a cycle in a directed graph.
15. Given knight's moves on a chessboard, find the minimum steps from A to B.
16. Mark all reachable cells in a grid from a given source cell.
17. Given a grid with walls (#) and empty cells (.), fill all reachable . from source with +.
18. Find the number of "blobs" (connected groups of 1s) in a binary image.
19. From a set of cities and roads, check if the network is fully connected.
20. Count the number of articulation points (cut vertices) in a small undirected graph.
21. Given a grid with teleportation cells, find the shortest time to reach the destination.
22. Given a $n \times m$ grid, flood-fill the area around a point.
23. Reconstruct the path from source to target using BFS.
24. Convert a tree into a rooted tree and store depth of all nodes using DFS.
25. Check if a path exists between two cities using adjacency list.
26. Given a grid, check if a cycle exists by moving up/down/left/right.
27. Use BFS to simulate water spreading over a terrain and count time taken to flood.
28. Find the number of leaf nodes in a tree represented with edges.
29. Perform topological sorting using DFS.
30. Given a fire spreading in a grid, check if you can escape before it reaches you.

These problems focus on **traversal**, **component discovery**, **reachability**, **path-finding**, and **cycle detection**, often on **small constraints** ($n \leq 1000$ or even $n \leq 100$) to allow DFS/BFS solutions.

Need solutions, visualizations, or problem links for any of these? Just ask!

You said:

Math (Mod, GCD, LCM, Divisors, Primes) Appears often in constraints and combinatorics.

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** focused on **Math concepts** like **Modulo, GCD, LCM, Divisors, Primes**, which appear frequently in **constraints, number theory, and combinatorics**:

÷ Math (Mod, GCD, LCM, Divisors, Primes) – 30 Problem Ideas

1. Given a and b , find $\gcd(a, b)$ and $\text{lcm}(a, b)$.
 2. Count how many numbers from 1 to n are divisible by at least one of a, b, c .
 3. For a given n , find the number of divisors of n .
 4. Check if a number n is a prime.
 5. Find the number of integers x such that $1 \leq x \leq n$ and $\gcd(x, n) = 1$.
 6. Given a, b , and m , compute $(a^b) \bmod m$.
 7. Given a list of numbers, find the GCD of all elements.
 8. Count how many prime numbers are there between 1 and n .
 9. Find the smallest positive integer x such that $\text{lcm}(x, n) = m$, if possible.
 10. Given n , output all prime divisors of n .
 11. Given an array, find the number of pairs (i, j) such that $\gcd(a[i], a[j]) = 1$.
 12. Find all numbers less than n that are relatively prime to n .
 13. Given n , find the sum of all its divisors.
 14. Count the number of integers x such that $x! \% n = 0$.
 15. For a given n , find the largest prime $\leq n$.
 16. Check whether two numbers a and b are coprime.
 17. Given n , count the number of unordered pairs (i, j) such that $1 \leq i, j \leq n$ and $\gcd(i, j) = 1$.
 18. Find the smallest $x > 1$ such that $\gcd(x, n) = 1$.
 19. Count how many numbers in a range $[l, r]$ are divisible by k .
 20. Given a, b, m , solve for x such that $(a * x) \% m = b$.
 21. Determine whether $(a^b) \bmod m$ is even or odd.
 22. Find the least number x such that $x \equiv a \pmod{m}$ and $x \equiv b \pmod{n}$.
 23. For a given n , find how many numbers $\leq n$ have exactly k divisors.
 24. Given n , print the k -th prime number.
 25. Count the number of integers in range $[1, r]$ with exactly 2 distinct prime divisors.
 26. For an array $a[]$, compute the product of all elements modulo 10^9+7 .
 27. Find the number of ways to write n as a product of two positive integers.
 28. Given n , generate all prime numbers using the Sieve of Eratosthenes.
 29. Check if a^b divides $c!$.
 30. Given n and m , find the number of integers x such that $1 \leq x \leq n$ and $x \bmod m = 0$.
-

These problems frequently involve **modulo arithmetic**, **divisibility rules**, **prime factorization**, and **efficient computation under constraints**.

Would you like solutions, Sieve implementations, or Codeforces links for these problems?

You said:

✓ Bit Manipulation Subset generation, XOR problems, bit-count.

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** based on ✓ **Bit Manipulation** — covering **subset generation**, **XOR operations**, **bit-counting**, and more. These problems often involve **binary logic**, **performance tricks**, and **clever state encoding**.

□ Bit Manipulation – 30 Problem Ideas

🧩 Subset Generation

1. Generate all possible subsets of a given array of n elements.
2. Find the subset of array with maximum XOR value.
3. Count the number of subsets whose sum is even.
4. From given integers, count subsets with exactly k elements having even XOR.
5. Among all non-empty subsets, find the one with maximum number of set bits in its XOR.

✂ XOR Problems

6. Given an array $a[]$, find the maximum XOR of any two elements.
7. Find the number of pairs (i, j) such that $a[i] \oplus a[j] = x$.
8. Find the smallest positive integer x such that $x \oplus a = b$.
9. Given an array, partition it into two subsets such that XOR of each subset is 0.
10. Find the value of $(a[0] \oplus a[1] \oplus \dots \oplus a[n-1])$.
11. Check if it is possible to split an array into two parts with equal XOR.
12. Given array $a[]$, find an integer x such that XOR of all $a[i] \oplus x$ is minimized.
13. Given n integers, find triplets (i, j, k) such that $a[i] \oplus a[j] = a[k]$.
14. Calculate XOR of all numbers from L to R .
15. For n queries of type (a, b) , compute XOR of elements in range $[a, b]$.

□ Bit Count / Set Bits

16. Count number of 1's in binary representation of given n .
 17. Find how many numbers between 1 and N have exactly k set bits.
 18. Given an integer, turn off the rightmost set bit.
 19. For given number n , count positions where both n and $n-1$ have set bits.
 20. Find the next number greater than n with the same number of set bits.
 21. Among all numbers from 1 to n , find the one with maximum set bits.
 22. Given two integers, check if they differ by exactly one bit.
-

□ Bitmask DP / Tricks

23. Given a list of n tasks and their dependencies encoded in bits, find a valid order to perform all.
 24. You are allowed to flip at most k bits — maximize the value of result.
 25. You can perform AND, OR, XOR operations — reach a target number using minimal steps.
 26. Using bitmasking, count number of ways to assign n jobs to n people (each person has skill set as bitmask).
 27. Check if a given number is a power of 2.
 28. Given two numbers, count how many bits you need to change to convert one into another.
 29. Among all integers with k set bits, find the smallest greater than n .
 30. Given n switches and some constraints (on/off in pairs), count valid switch configurations.
-

These problems test your understanding of **bitwise operators** (&, |, ^, ~, <<, >>), and how to apply them efficiently in logic-heavy problems.

Would you like full Codeforces problem links, solutions, or C++/Java implementation examples?

You said:

Hashing (Maps, Sets) Frequency counts, unique tracking, hash-based memoization

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** involving **Hashing with Maps and Sets**, covering patterns like **frequency counting, uniqueness tracking, and hash-based memoization** — all key to optimizing time complexity in constraints-heavy problems.

□ Hashing (Maps, Sets) – 30 Problem Ideas

🔄 Frequency Count

1. Count the frequency of each element in an array.
 2. Given a string, find the character with the highest frequency.
 3. Given n numbers, find the number that appears more than $n/2$ times (if any).
 4. Determine if two strings are anagrams of each other.
 5. Count the number of pairs (i, j) with $i < j$ and $a[i] = a[j]$.
 6. Find the number of subarrays where frequency of element x is greater than y .
 7. For each element in an array, print how many times it has appeared before.
 8. Count the number of elements in the array with frequency exactly k .
 9. Given two arrays, count how many elements are common.
 10. Find the most frequent element in all sliding windows of size k .
-

✓ Unique Tracking

11. Find the first non-repeating character in a stream.
 12. Given a list of usernames, generate unique usernames by appending numbers if duplicates exist.
 13. Determine the number of distinct elements in every subarray of size k .
 14. Check if a given array has any duplicates.
 15. Find the number of unique substrings of a string.
 16. Count the number of unique elements that appear only once in the array.
 17. Given an array, determine the number of distinct absolute differences between pairs.
 18. Given a string, find the length of the longest substring with all unique characters.
 19. Track the first occurrence index of each unique element in an array.
 20. Count the number of distinct prime factors among n numbers.
-

📦 Hash-Based Memoization / Hashing Tricks

21. Use a map to store prefix sums and count the number of subarrays with sum k .
 22. Memoize results of a recursive function using a hash map to avoid recomputation.
 23. Count the number of subarrays whose XOR is equal to x using prefix XOR + hashing.
 24. Store previously seen states of a grid-based game to detect cycles.
 25. Hash string representations of trees to check for subtree duplication.
 26. Given multiple strings, group them by anagram class.
 27. Use `unordered_set` to store all substrings of length k for quick duplicate check.
 28. Store previously visited nodes in a BFS using hash set to prevent revisits.
 29. In a recursive game, store game states as hashable tuples to avoid TLE.
 30. Given a 2D matrix, hash all row combinations to count duplicates.
-

These problems are classic use-cases for `unordered_map`, `unordered_set`, or `HashMap/HashSet` in Java — enabling **$O(1)$ lookups**, **duplicate detection**, and **memoization** to handle large input sizes efficiently.

Would you like categorized real Codeforces problems, or sample solutions for these?

You said:

Sliding Window Substring, subarray optimizations.

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** involving the **Sliding Window** technique — ideal for optimizing **substring/subarray problems**, especially when dealing with continuous segments and fixed/moving ranges.

Sliding Window – 30 Problem Ideas

Substring / String Sliding Window

1. Find the length of the longest substring with all unique characters.
 2. Given a string, find the minimum window that contains all characters of another string.
 3. Count the number of substrings with exactly k distinct characters.
 4. Find the longest substring where no character appears more than k times.
 5. Given a binary string, find the length of the longest substring with at most k zeros.
 6. Find the number of substrings with at most k different vowels.
 7. Count the number of substrings with sum of ASCII values $\leq x$.
 8. Given a string, find the smallest substring with equal number of a , b , and c .
 9. Replace at most k characters in a string to get the longest repeating character substring.
 10. Given two strings, find the maximum length of common substring using sliding window.
-

Subarray (Array-based Sliding Window)

11. Find the length of the longest subarray with sum $\leq k$.
12. Count the number of subarrays of length k with average $\geq x$.
13. Find the maximum sum of a subarray of size k .
14. Find the minimum size subarray whose sum is $\geq \text{target}$.
15. Count the number of subarrays where all elements are distinct.
16. Find the number of subarrays with at most k odd numbers.
17. Find the number of subarrays with at most k inversions.
18. Find the number of subarrays with sum equal to x using prefix sum + sliding window.
19. Determine the length of the longest subarray with equal number of 0s and 1s.
20. Find the longest subarray with at most two distinct numbers.

▣ Variants and Optimization Use-Cases

21. Find the minimum number of swaps to bring all 1s together in a binary array.
22. Given a circular array, find the maximum sum of any window of size k .
23. Find the maximum difference between the max and min in any subarray of size k .
24. Track max element in each subarray of size k using deque (monotonic queue).
25. Count how many windows of size k have strictly increasing elements.
26. Given energy levels, find the maximum time you can keep consuming energy $\leq x$.
27. Find the length of the longest subarray with sum divisible by k .
28. Find the maximum number of consecutive days you can play video games given rest rules.
29. Find the minimum length of a subarray you must remove to make the remaining array sorted.
30. Given a list of events with start and end days, find the maximum number of non-overlapping events in a k -day window.

Sliding Window helps bring brute force $O(n^2)$ solutions down to $O(n)$ or $O(n \log k)$ with extra data structures (like heaps or deques). Common in **interviews and time-limited contests**.

Would you like full problems with constraints or solutions using Python/Java/C++?

You said:

Simple Dynamic Programming (1D/2D) E.g., LIS, Knapsack (small constraints), LCS base cases

ChatGPT said:

Here are **30 one-line Codeforces-style problem statements** that use **Simple Dynamic Programming (1D/2D)** — ideal for **beginner to intermediate level** problems involving **LIS, Knapsack, LCS**, and similar patterns with **small constraints**.

☐ Simple Dynamic Programming (1D/2D) – 30 Problem Ideas

📌 1D DP Problems

1. Given n , find the number of ways to climb stairs taking 1 or 2 steps at a time.
2. Find the length of the longest increasing subsequence (LIS) in an array.
3. Given n , find the minimum number of squares that sum to n (Perfect Squares DP).
4. Count the number of binary strings of length n without consecutive 1s.
5. Given a target sum s , determine if any subset of array $a[]$ sums to s (Subset Sum).

6. Given n coins and amount x , find the number of ways to make sum x .
 7. Find the minimum number of coins needed to make sum x using given denominations.
 8. Given string s , count the number of ways to decode it using 1–26 letter codes.
 9. Given n , compute the n th Fibonacci number using memoization.
 10. Given a rod of length n and prices for each length, find the maximum profit (Rod Cutting).
-

Knapsack-Type DP (0/1 or Unbounded)

11. Given items with weight and value, and total capacity w , find the max value (0/1 Knapsack).
 12. Given n items and unlimited supply, find max value under weight limit (Unbounded Knapsack).
 13. You can use each item at most k times — maximize total value under weight limit.
 14. Given n books with cost and pages, maximize pages you can buy with given money.
 15. Split array into two subsets with minimum absolute difference of their sums.
-

2D DP Problems (LCS / Grid)

16. Find the length of the longest common subsequence (LCS) between two strings.
 17. Find the edit distance between two strings using insert/delete/replace.
 18. Count the number of unique paths from top-left to bottom-right in a grid.
 19. Given a grid with costs, find the minimum cost path to bottom-right corner.
 20. Count the number of ways to reach a cell (i, j) moving only right and down.
 21. Find the length of the longest palindromic subsequence in a string.
 22. Find the minimum number of insertions to make a string a palindrome.
 23. Given a matrix of 0s and 1s, find the largest square of 1s.
 24. Compute number of ways to make change for amount x using n coins.
 25. Count the number of distinct subsequences of a string that match another string.
-

Miscellaneous Easy-Medium DPs

26. Count the number of ways to tile a $2 \times n$ board using 1×2 tiles.
 27. Count how many arrays of size n have sum s with values from 1 to k .
 28. Given a string, find the minimum number of cuts to partition it into palindromic substrings.
 29. Count the number of binary trees that can be made with n nodes (Catalan DP).
 30. You are at step 0, and can jump from 1 to k steps — count the number of ways to reach step n .
-

These are **classic DP building blocks**, often used as subroutines in **more complex problems** later. Constraints are usually small ($n \leq 1000$ or less), making them ideal for **learning table-based DP**.

Need categorized solutions, visual diagrams, or Python/C++ implementations? Let me know!