Implementation one line rating 0 to 1199 problem statement in one line

ChatGPT said:

Here are **one-line problem statements** focused on the **"Implementation"** topic suitable for **Codeforces rating 0 to 1199**:

1. Simulate a basic calculator that supports `+`, `-`, `*`, `/`.
2. Swap two numbers without using a third variable.
3. Count digits in a given number.
4. Check if a string is a palindrome.
5. Reverse a given string.
6. Rotate an array by one position.
7. Count vowels and consonants in a string.
8. Find the maximum of three numbers.
9. Print numbers from 1 to N without a loop (using recursion).
10. Simulate a simple voting system and declare the winner.
11. Count how many times a substring appears in a string.
12. Replace all spaces in a string with hyphens.
13. Simulate elevator movement between floors based on input.
14. Simulate a digital clock till a given time.
15. Sort three integers using only if-else conditions.
16. Convert a string to lowercase without built-in function.
17. Check whether parentheses in a string are balanced.
18. Convert binary to decimal and vice versa.
19. Toggle case of each character in a string.
20. Extract digits from a string.
21. Simulate a traffic light system using conditions.
22. Implement a basic Caesar Cipher encryption.
23. Count the number of steps to reach 1 using Collatz sequence.
24. Implement a simple password strength checker.
25. Find the second largest element in an array.
26. Flip all bits in a binary string.
27. Replace duplicate characters with _ in a string.
28. Merge two sorted arrays into one.
29. Determine if a number is “almost prime” (has exactly two prime divisors).
30. Reorder characters in a string alphabetically.
31. Count how many times a digit occurs in a range.
32. Check if time is valid (HH:MM format).
33. Simulate bank account operations (deposit/withdraw).
34. Compress a string using run-length encoding.
35. Find the first non-repeating character in a string.
36. Simulate an ATM machine with limited notes.
37. Implement left and right circular shift of a string.
38. Create a frequency table of elements in an array.
39. Print diagonals of a 2D matrix.
40. Simulate snake movement on a grid.
41. Format a number with commas (e.g., 1000000 → 1,000,000).
42. Remove all adjacent duplicates in a string.
43. Simulate basic typing with CapsLock toggle.

44. Implement multiplication of two matrices.
45. Find row with maximum number of 1s in a binary matrix.
46. Simulate queue operations (enqueue, dequeue).
47. Count number of bits set in an integer.
48. Check if a sequence is strictly increasing or decreasing.
49. Rotate a matrix 90 degrees clockwise.
50. Simulate a voting system with tie-breaking rule.

Let me know if you want **solutions**, **grouped by pattern** (like simulation, string, matrix, etc.), or **next level (1200–1399)**.

You said:

Math (Basic Number Theory) GCD, LCM, modulo, primes, divisors. 30 problem in one line

ChatGPT said:

Here are **30 one-line problem statements** on **Basic Number Theory** (GCD, LCM, modulo, primes, divisors) suitable for **Codeforces rating 0 to 1199**:

□ GCD & LCM

1. Find the GCD of two given numbers.
2. Compute LCM of two given integers.
3. Calculate GCD of an array of integers.
4. Compute LCM of an array of numbers.
5. Count the number of pairs in an array with GCD equal to 1.
6. Check if two numbers are co-prime.
7. Find smallest number divisible by both A and B.
8. Print all pairs (i, j) such that GCD(i, j) = K.

□ Modulo Arithmetic

9. Compute `(A × B) % M` safely for large A and B.
10. Check if `(A + B) % M == (A % M + B % M) % M`.
11. Compute `(A^B) % M` using modular exponentiation.
12. Solve `(A * x) % M = 1` to find modular inverse.
13. Calculate `(A - B + M) % M` to handle negative modulo.
14. Find remainder when a large number (as string) is divided by M.
15. Evaluate sum of first N natural numbers modulo M.

🔢 Prime Numbers

16. Check if a number is prime.

17. Print all primes up to N (Sieve of Eratosthenes).
18. Count number of primes in range [L, R].
19. Find next prime number greater than given N.
20. Count number of twin prime pairs up to N.
21. Generate first N prime numbers.
22. Check if a number is a product of exactly two primes.

🔍 Divisors & Factors

23. Print all divisors of a number in sorted order.
24. Count total number of divisors of a number.
25. Sum all divisors of a number.
26. Check if a number is a perfect square and its square root is an integer.
27. Count number of common divisors of two numbers.
28. Count number of divisors of N that are even.
29. Count number of numbers in range [1, N] divisible by A or B.
30. Check whether a number is a powerful number (all prime factors have exponent ≥ 2).

Let me know if you want solutions or want to continue with **Math (Digits, Sums, Counting)** problems next.

You said:

greddy Algo

ChatGPT said:

Here are **30 one-line problem statements** focused on **Greedy Algorithms**, ideal for **Codeforces rating 0 to 1199**:

✅ Greedy Algorithms – One-line Problems

1. Choose minimum number of coins to make a given sum.
2. Select maximum number of non-overlapping intervals from given start and end times.
3. Choose minimum number of platforms needed so no train waits.
4. Schedule jobs to maximize total profit with deadlines.
5. Find minimum number of jumps to reach end of array.
6. Minimize sum of absolute differences between elements of two arrays.
7. Maximize total value by selecting items with weight ≤ W (fractional knapsack).
8. Choose minimum number of operations to make all array elements equal.
9. Divide N chocolates among friends to maximize minimum share.
10. Place minimum number of cameras to monitor all nodes in a binary tree.
11. Buy products to minimize cost when price drops every day.
12. Choose the maximum number of activities you can do (activity selection).

13. Maximize number of meetings in a single room without overlap.
14. Choose digits from a number to create the largest number of length K.
15. Arrange people in queue to minimize total waiting time.
16. Distribute candies to children with conditions on ratings.
17. Fill containers with weights to minimize number of containers used.
18. Remove K digits from a number to make it the smallest possible.
19. Choose intervals to cover a full segment with minimal number of intervals.
20. Find smallest lexicographical string by removing K characters.
21. Pick students to maximize total marks without picking adjacent ones.
22. Cut minimum number of ropes to make all equal length.
23. Choose maximum number of tasks from a queue before deadline.
24. Buy products to maximize savings using discounts in order.
25. Assign rooms to guests to minimize number of rooms used.
26. Reorganize a binary string to get the lexicographically smallest one.
27. Choose team members with highest skill under budget limit.
28. Place the maximum number of cows in stalls with minimum distance.
29. Distribute pages to students to minimize maximum pages assigned.
30. Select balloons with highest score within time limit.

Let me know if you want:

- **Grouped by pattern** (intervals, coins, strings, etc.)
- **Solutions or hints**
- **Next difficulty level (1200–1399)**

You said:

Brut force

ChatGPT said:

Here are **30 one-line problem statements** focused on the **Brute Force** technique, suitable for **Codeforces rating 0 to 1199**:

□ Brute Force – One-Line Problems

1. Try all pairs in an array to find the one with the given sum.
2. Generate all substrings of a string and print the longest palindrome.
3. Find the maximum product of any three elements in an array.
4. Count number of triplets that sum to zero.
5. Check all permutations of a string to see if any is a palindrome.
6. Generate all combinations of size K from 1 to N.
7. Count how many permutations of digits form a number divisible by 3.
8. Find the smallest missing positive integer by checking each one.
9. Try all substrings to find the one with maximum number of vowels.
10. Count pairs (i, j) such that i < j and A[i] > A[j].

11. Try all possible partitions to check if a string can be split into palindromes.
12. Generate all subsets of an array and find the one with maximum sum ≤ K.
13. Try every possible hour and minute to find the next palindromic time.
14. For all subarrays, find the one with the minimum sum.
15. Try all possible ways to arrange N rooks on an NxN board.
16. Check all numbers from 1 to N for being Armstrong numbers.
17. Try all pairs of indices to find the minimum distance between same elements.
18. Generate all strings of length N using lowercase letters.
19. For each pair (i, j), check if A[i] is divisible by A[j].
20. Try all pairs of points to find the smallest Euclidean distance.
21. Count all distinct substrings in a string using a set.
22. Check if array can be sorted by reversing any one subarray.
23. Try all possible bitmasks to find the maximum XOR subset.
24. Generate all binary strings of length N with no consecutive 1s.
25. Try all number combinations of 4 digits to match a password.
26. Simulate all possible moves in a tic-tac-toe game.
27. Check every divisor of each number in the array.
28. Try all paths from (0, 0) to (N, M) in a grid (without optimization).
29. Rotate array all possible ways and check for sorted version.
30. Count all palindromic numbers between L and R.

Let me know if you'd like:

- **Grouped patterns (strings, arrays, math, etc.)**
- **Next difficulty range (1200–1399)**
- **Solutions in C++ or Java or Python**

You said:

Two Pointers For string and array subrange problems

ChatGPT said:

Here are **30 one-line problem statements** using the **Two Pointers** technique, focused on **string and array subrange problems**, ideal for **Codeforces rating 0–1199**:

🔁 Two Pointers – Arrays & Strings

✅*Arrays:*

1. Find a pair in a sorted array with a given sum.
2. Count number of pairs in array with sum less than K.
3. Remove duplicates from a sorted array in-place.
4. Merge two sorted arrays into one sorted array.
5. Find subarray with given sum in non-negative array.
6. Count subarrays with at most K distinct elements.

7. Find longest subarray with all distinct elements.
8. Find smallest subarray with sum ≥ S.
9. Move all zeroes to end of the array without using extra space.
10. Partition array into two parts with equal sum.
11. Find max length of subarray with sum exactly K.
12. Count number of subarrays with even sum.
13. Count number of subarrays with product less than K.
14. Check if array has a subarray which is a permutation of another.
15. Rotate array using reverse + two pointers.

✅ *Strings:*

16. Find the longest substring with all unique characters.
17. Count number of substrings with at most K distinct characters.
18. Find smallest window in a string containing all characters of another string.
19. Check if one string is a permutation of another substring.
20. Remove all duplicate characters from string in-place.
21. Check if string can be converted into palindrome by removing one character.
22. Compress string by grouping repeated characters (like run-length encoding).
23. Compare two strings with backspaces (`#`) using two pointers.
24. Find longest common prefix of two strings.
25. Find if a string is a rotation of another using pointers.
26. Find minimum number of characters to remove to make two strings equal.
27. Search for pattern in text using sliding window.
28. Check if a string has a palindrome substring of length K.
29. Find longest repeating character replacement with at most K changes.
30. Trim leading and trailing spaces using two pointers.

Let me know if you want:

- Problems grouped by **difficulty**
- **Hints or full solutions** in your preferred language (Java, C++, Python)
- Continue to **next technique: Prefix Sum / Sliding Window**

You said:

Prefix Sum / Difference Arrays For range sum or difference queries

ChatGPT said:

Here are **30 one-line problem statements** based on **Prefix Sum / Difference Arrays**, especially for **range sum or difference queries**, suited for **Codeforces rating 0–1199**:

🔢 Prefix Sum – Arrays (Range Queries)

✓ *Basic Prefix Sum:*

1. Compute the sum of elements from index `L` to `R` in an array.
2. Find the number of even numbers in range `[L, R]`.
3. Count how many times a value appears in a given range.
4. Find the average of elements between indices `L` and `R`.
5. Check if sum of range `[L, R]` is divisible by `K`.
6. Precompute prefix sum to answer multiple range sum queries.
7. Find the maximum sum subarray ending at each index.
8. Find the difference between the sum of even and odd indexed elements.
9. Count how many elements in range `[L, R]` are greater than X.
10. Calculate frequency of digits 0–9 in a large numeric string prefix.

✓ *2D Prefix Sum:*

11. Find the sum of a submatrix given its top-left and bottom-right corners.
12. Count the number of 1s in a rectangular area of a binary matrix.
13. Compute prefix XOR for submatrices.
14. Find the sum of border elements in a submatrix.
15. Check if all elements in submatrix are the same.

✓ *Difference Array (Efficient Range Updates):*

16. Add value `X` to a subarray from index `L` to `R`.
17. Increment elements in a range `[L, R]` multiple times using a difference array.
18. Apply multiple range update operations and find final array.
19. Perform a range addition, then calculate prefix sum of result.
20. Simulate N operations of adding values to ranges efficiently.

✓ *Advanced Prefix Sum Uses:*

21. Find the longest subarray with sum less than or equal to K.
22. Count subarrays with sum exactly equal to K.
23. Find if there exists a subarray of length at least K with positive sum.
24. Count number of subarrays where sum is even.
25. Count number of subarrays where prefix sum is divisible by M.
26. Check if prefix sum equals suffix sum at any index.
27. Detect zero-sum subarray using prefix sum and hashing.
28. Count the number of range sums that lie in [lower, upper].
29. Find smallest prefix whose sum is at least X.
30. Compute cumulative XOR for all prefixes of an array.

Let me know if you'd like:

- Problems **grouped by usage (sum, frequency, updates)**
- Solutions in **Java / C++ / Python**
- Next topic: **Counting Techniques / Hashing / Sliding Window**

2/2

You said:

✅ Arrays and Strings Core data types used in almost every problem

ChatGPT said:

Here are **30 one-line problem statements** based on **Arrays and Strings**, the **core data types** used in almost every problem—perfect for **Codeforces rating 0–1199**:

🔢 Array Problems

1. Reverse an array in-place.
2. Find the maximum element in an array.
3. Find the second smallest element in an array.
4. Count frequency of each element in an array.
5. Rotate array to the right by K positions.
6. Merge two sorted arrays into one sorted array.
7. Remove duplicates from a sorted array.
8. Move all zeroes to the end of the array.
9. Count pairs whose sum is equal to a target.
10. Check if array contains duplicate elements.
11. Find common elements in two arrays.
12. Check if array is sorted in ascending order.
13. Return indices of two numbers adding up to a target.
14. Find the subarray with the maximum sum (Kadane's algorithm).
15. Replace each element with the greatest element to its right.

🔠 String Problems

16. Reverse a string in-place.
17. Check if a string is a palindrome.
18. Count vowels and consonants in a string.
19. Remove all duplicate characters from a string.
20. Convert a string to lowercase without using built-in function.
21. Find the first non-repeating character in a string.
22. Check if two strings are anagrams of each other.
23. Compress a string using run-length encoding.
24. Count the frequency of each character in a string.

25. Find the longest common prefix in an array of strings.
26. Replace all spaces in a string with '%20'.
27. Check if string A is a rotation of string B.
28. Count number of words in a sentence.
29. Find the longest substring without repeating characters.
30. Remove all digits from a string.

Let me know if you'd like:

- ✅ Solutions in Java / C++ / Python
- ✅ Categorized by pattern (search, sort, transform, count)
- ✅ More problems at **intermediate level (1200–1399)**

You said:

Sorting + Binary Search (Basic) Key to many greedy and search problems

ChatGPT said:

Here are **30 one-line problem statements** on **Sorting + Binary Search (Basic)** — essential techniques for solving many **Greedy and Search** problems, especially for **Codeforces rating 0–1199**:

🔄 Sorting – Basic Problems

1. Sort an array of integers in ascending order.
2. Sort strings in lexicographical (dictionary) order.
3. Sort pairs by first element and then by second.
4. Sort array by frequency of elements (high to low).
5. Sort students by marks (descending) and name (ascending).
6. Sort an array and remove duplicates.
7. Sort an array of 0s, 1s, and 2s (Dutch National Flag problem).
8. Sort words in a sentence by length.
9. Sort an array such that all even numbers come before odd.
10. Sort intervals by start time, then merge overlapping ones.
11. Sort points based on their distance from origin.
12. Sort strings by number of vowels.
13. Sort array and return the index of the K-th smallest element.
14. Sort a list of fractions in increasing order.
15. Sort characters in a string by frequency.

🔎 Binary Search – Basic Applications

16. Find an element in a sorted array using binary search.
17. Find the first occurrence of a number in a sorted array.
18. Find the last occurrence of a number in a sorted array.
19. Count how many times a number occurs in a sorted array.
20. Find square root of a number using binary search.
21. Find the smallest element greater than or equal to a target.
22. Find the largest element smaller than or equal to a target.
23. Check if a number is a perfect square using binary search.
24. Search for an element in a rotated sorted array.
25. Find the minimum in a rotated sorted array.
26. Find peak element in a mountain array.
27. Find the K-th smallest element in a sorted matrix.
28. Use binary search to find the number of elements ≤ X.
29. Find median of two sorted arrays (naive two-pointer + binary search).
30. Find the smallest positive number that is missing in an unsorted array.

Let me know if you’d like:

- ✅ **Solutions** in C++ / Java / Python
- ✅ Split by **Greedy usage** or **Search optimizations**
- ✅ Continue to **next topic**: Graphs / DP / Hashing etc.

You said:

Maps/Sets (Hashing) Counting elements, uniqueness, frequency.

ChatGPT said:

Here are **30 one-line problem statements** based on **Maps/Sets (Hashing)** — especially for **counting elements, uniqueness, and frequency**, perfect for **Codeforces rating 0–1199**:

📊 Maps / Hash Maps – Frequency & Counting

1. Count the frequency of each element in an array.
2. Count how many times each character appears in a string.
3. Find the element with the highest frequency in an array.
4. Find the first non-repeating character in a string.
5. Count the number of distinct elements in an array.
6. Find duplicate elements in an array using a map.
7. Find the most frequent word in a sentence.
8. Group words that are anagrams using a hash map.
9. Count frequency of all prefixes of a string.
10. Find top K frequent elements in an array.
11. Count the number of subarrays with sum equal to K.
12. Check if two strings are anagrams using maps.
13. Find common elements between two arrays using maps.

14. Count number of elements with frequency greater than 1.
15. Track the number of times each query appears in a list.

✅ Sets / Hash Sets – Uniqueness & Lookup

16. Check if all elements in an array are unique.
17. Remove duplicates from an array using a set.
18. Check if a string has all unique characters.
19. Store all substrings of a string in a set.
20. Count number of distinct substrings in a string.
21. Check if there exists a pair with a given sum in an array.
22. Count number of distinct absolute values in an array.
23. Find the intersection of two arrays using sets.
24. Check if any value occurs at least twice in an array.
25. Store and print only unique words from a sentence.
26. Find the smallest missing positive number using set.
27. Determine if a subarray with zero sum exists.
28. Count number of distinct elements in every window of size K.
29. Check if any permutation of a string is a palindrome (using odd count check).
30. Count the number of distinct characters in every substring of length K.

Let me know if you'd like:

- ✅ C++ / Java / Python **solutions**
- ✅ Grouped by **frequency**, **uniqueness**, or **set operations**
- ✅ Move to next topic: **Sliding Window**, **Graphs**, or **Dynamic Programming**

2/2

You said:

Basic Recursion / Backtracking Subset generation, small dfs-style problems

ChatGPT said:

Here are **30 one-line problem statements** on **Basic Recursion and Backtracking**, covering **subset generation, permutations, and DFS-style problems**, ideal for **Codeforces rating 0–1199**:

🔁 Basic Recursion Problems

1. Print numbers from 1 to N using recursion.
2. Print elements of an array in reverse using recursion.

3. Compute factorial of a number using recursion.
4. Find nth Fibonacci number using recursion.
5. Calculate power of a number (A^B) using recursion.
6. Find GCD of two numbers using recursive Euclidean algorithm.
7. Print sum of digits of a number using recursion.
8. Count number of digits in a number using recursion.
9. Reverse a string using recursion.
10. Check if a string is palindrome using recursion.
11. Find maximum element in an array using recursion.
12. Count occurrences of a character in a string recursively.
13. Print all even numbers from N to 1 using recursion.
14. Convert decimal to binary using recursion.
15. Generate nth term of the Tribonacci sequence using recursion.

□ Backtracking / DFS-style Problems

16. Generate all subsets (power set) of an array.
17. Generate all permutations of a string.
18. Solve N-Queens for N = 4 using backtracking.
19. Print all paths from top-left to bottom-right in an MxN grid.
20. Generate all combinations of K numbers from 1 to N.
21. Solve the Rat in a Maze problem (binary matrix paths).
22. Generate all binary strings of length N with no consecutive 1s.
23. Print all palindromic partitions of a string.
24. Solve Sudoku using backtracking (simplified 4x4).
25. Generate valid parentheses combinations for N pairs.
26. Find if there's a path in a grid avoiding obstacles using DFS.
27. Place K knights on a chessboard without attacking each other.
28. Generate all possible phone number combinations from digit mapping.
29. Generate all valid subsets with sum ≤ K.
30. Backtrack to remove minimum characters to make a string palindrome.

Let me know if you'd like:

- ✅ Solutions or step-by-step recursive trees
- ✅ Java / C++ / Python implementations
- ✅ Move to **Dynamic Programming**, **Graphs**, or **Advanced Backtracking** next

2/2

You said:

Max/Min subarray, subsegment Array/sorting/prefix sum-based

ChatGPT said:

Here are **30 one-line problem statements** focused on **Max/Min Subarray/Subsegment** problems using **Array, Sorting, Prefix Sum**, suitable for **Codeforces rating 0–1199**:

▲ Maximum Subarray / Subsegment

1. Find the maximum sum subarray (Kadane's algorithm).
2. Find the length of the longest subarray with sum ≤ K.
3. Find the subarray with maximum average.
4. Find the maximum sum of K consecutive elements.
5. Find maximum sum subarray of size K.
6. Find the longest subarray with all positive numbers.
7. Find maximum subarray sum in circular array.
8. Find subarray with maximum product.
9. Find subarray with maximum XOR value.
10. Find longest subsegment with sum divisible by K.
11. Maximize sum of any subarray after at most one reversal.
12. Find maximum difference between sum of two non-overlapping subarrays.
13. Count number of subarrays with sum equal to maximum subarray sum.
14. Find maximum possible sum after replacing at most one element with K.
15. Maximize score of subarray where score = min * length.

▼ Minimum Subarray / Subsegment

16. Find subarray with minimum sum.
17. Find the shortest subarray with sum ≥ K.
18. Find minimum sum of K consecutive elements.
19. Find smallest subsegment with all distinct elements.
20. Find subarray with minimum positive sum.
21. Find shortest subarray to remove so array becomes sorted.
22. Find subarray with minimum product.
23. Find minimum length subarray with exactly K distinct elements.
24. Remove one element to minimize maximum subarray sum.
25. Find minimum difference between sums of two subarrays of same size.
26. Find subarray whose sum is minimum but contains at least one even number.
27. Find shortest subsegment with sum divisible by X.
28. Find minimum prefix to remove to make subarray sum divisible by D.
29. Find minimum subarray length that contains both max and min elements.
30. Find subarray with minimum XOR value.

Let me know if you'd like:

- ✅ Solutions in C++ / Java / Python
- ✅ Grouped by algorithm (Kadane's, prefix sum, sliding window)

- ✅ Move to **Sliding Window**, **Greedy**, or **DP-based optimization** problems next

You said:

Frequency Count

ChatGPT said:

Here are **30 one-line problem statements** based on **Frequency Count** – essential for problems involving **hashing, maps, arrays, strings**, perfect for **Codeforces rating 0–1199**:

🔢 Frequency Count – Arrays

1. Count frequency of each element in an array.
2. Find the most frequent element in an array.
3. Find elements that appear exactly once.
4. Find elements that appear more than ⌊n/2⌋ times (Majority Element).
5. Check if two arrays are equal (same frequency of elements).
6. Find all elements that appear more than n/3 times.
7. Count frequency of elements in range [L, R].
8. Identify duplicate elements in an array.
9. Check if frequency of all elements is unique.
10. Find the element with second-highest frequency.

🔤 Frequency Count – Strings

11. Count frequency of each character in a string.
12. Find the most frequent character in a string.
13. Check if two strings are anagrams (frequency match).
14. Count frequency of each word in a sentence.
15. Count the number of palindromic words in a sentence.
16. Check if a string can be rearranged to form a palindrome.
17. Find first non-repeating character in a string.
18. Remove characters that appear more than once.
19. Compare character frequency between two strings.
20. Print characters sorted by frequency in descending order.

Combined & Advanced Frequency Uses

21. Count number of elements with frequency > 1.
22. Count number of subarrays with same frequency of 0s and 1s.
23. Check if frequency of digits in a number is balanced.
24. Use frequency array to find missing element from second array.

25. Count frequency of remainders when array elements are divided by K.
26. Use frequency map to find K most frequent elements.
27. Track and update frequency of elements dynamically (add/remove).
28. Build prefix frequency of characters in a string.
29. Check if all characters in a string occur same number of times.
30. Determine if two strings are isomorphic (same frequency pattern mapping).

Let me know if you want:

- ✅ Solutions in Java / C++ / Python
- ✅ Grouping by **strings**, **arrays**, or **applications (palindromes, anagrams)**
- ✅ Next topic: **Sliding Window**, **Hash Maps + Logic**, or **Two Pointers + Frequency**

You said:

Greedy sorting Sort + simulate (smallest/largest first)

ChatGPT said:

Here are **30 one-line problem statements** for **Greedy + Sorting** (i.e. **Sort + Simulate**, picking **smallest/largest first**) — ideal for **Codeforces rating 0–1199**:

☐ Greedy + Sorting (Sort & Simulate)

▲ Pick Smallest First (Minimize)

1. Select the minimum number of coins to form a given amount.
2. Assign minimum number of platforms for trains based on arrival/departure times.
3. Find the minimum number of arrows to burst all balloons (sorted intervals).
4. Minimize the cost to connect ropes into one (Huffman-style merge).
5. Choose minimum number of meetings to attend to cover all slots.
6. Distribute candies to minimize unfairness (minimize max - min in subarray).
7. Distribute tasks to two workers to minimize max total time.
8. Choose minimum number of jumps to reach end of array.
9. Find the minimum sum of absolute differences of two arrays.
10. Assign jobs to workers to minimize maximum individual time.

▼ Pick Largest First (Maximize)

11. Choose activities to attend to maximize total reward.
12. Maximize number of events attended without overlap.
13. Maximize total value in fractional knapsack problem.
14. Maximize number of toys bought within budget.
15. Pick projects with max profit while staying within time limit.

16. Select players to maximize team strength within salary cap.
17. Select meetings with max duration without overlap.
18. Maximize tips from customers by picking highest tip first.
19. Choose tasks to complete to get max marks within given deadline.
20. Choose segments to cover a range with minimal number of cuts.

▯ Sort + Simulate Scenarios

21. Sort children and cookies to distribute cookies to most kids.
22. Sort items by value-to-weight ratio for fractional knapsack.
23. Simulate order of car fueling to minimize refills.
24. Schedule jobs to meet deadlines and maximize reward.
25. Sort files by size and simulate compression order.
26. Simulate order of painting walls with drying time constraint.
27. Minimize total penalty by doing shortest tasks first.
28. Sort balloons by size and assign to people to maximize use.
29. Sort boxes and trucks to fit maximum boxes in trucks.
30. Simulate greedy player turns picking highest value coins from ends.

Let me know if you'd like:

- ✅ Java/C++/Python **solutions**
- ✅ Categorization (minimize cost, maximize profit, scheduling)
- ✅ Move to **Advanced Greedy**, **Sweep Line**, or **Graph-based Greedy (Prim/Kruskal)**

You said:

Brute Force search Try all possibilities

ChatGPT said:

Here are **30 one-line problem statements** for the **Brute Force (Try All Possibilities)** approach — especially useful for **Codeforces 0–1199 level**:

🔍 Brute Force — Try All Possibilities

✅Subsets, Permutations, and Combinations

1. Print all subsets of a given set of numbers.
2. Count the number of subsets that sum to a target.
3. Generate all permutations of a string or array.
4. Find the lexicographically next permutation of an array.
5. Count all valid parenthesis combinations of `n` pairs.

6. Check if a string is an anagram of any substring of another string.
7. Count the number of ways to climb stairs with 1 or 2 steps.
8. Print all binary strings of length `n` with no two consecutive 1's.
9. Generate all palindromic substrings of a string.
10. Generate all possible outcomes of a dice throw (1–6) of `n` dice.

⍰ Try All Index Pairs / Triplets / Paths

11. Check if any two elements in array sum to a target.
12. Count all triplets `(i, j, k)` such that `arr[i] + arr[j] + arr[k] == 0`.
13. Find maximum sum of any subarray (brute-force all ranges).
14. Count the number of inversions in an array.
15. Check if a number can be written as the sum of three primes.
16. Find the rectangle in a grid with maximum sum (try all corners).
17. Count all squares or rectangles in a binary matrix.
18. Try all permutations of visiting `n` cities (TSP-style brute-force).
19. Check all pairs of strings to find if any are rotations of each other.
20. Check if any rotation of string `A` is equal to string `B`.

⍰ Brute-Force Simulation / Decision Making

21. Try placing `k` knights on `n x n` board so no two attack each other.
22. Try all arrangements of people in a line and find the valid ones.
23. Simulate every second of a race to check for overtake moments.
24. Brute force through all ways to cut a rope into 3 parts with max product.
25. Find the number of ways to reach a cell from top-left in a grid.
26. Try all positions to place a mirror to reflect a laser to the target.
27. Try all string reversals to see which gives palindrome.
28. Brute force all possibilities of putting signs `+` or `-` to reach target sum.
29. Try all positions to split a string to make both halves palindromes.
30. Try all possible swaps to make an array sorted with minimum moves.

Would you like:

- Java or C++ solutions for some of these?
- Categorized brute force templates?
- Practice ladder links (Codeforces, AtCoder, etc.)?

Let me know!

You said:

Small Graph Problems (Optional) BFS/DFS on small grids (for 1100-1200

ChatGPT said:

Here are **30 small graph problems (1100–1200 level)** focusing on **BFS/DFS on small grids**, especially helpful for Codeforces & beginner contests.

🔁 BFS / DFS on Small Grids (1100–1200)

Basic Grid Traversal

1. Count number of islands (connected 1s in 2D grid).
2. Flood fill a region in a grid starting from a given cell.
3. Mark all boundary-connected 0s in a binary matrix.
4. Find the shortest path in a grid with only right and down moves.
5. Count connected components in a binary matrix.
6. Determine if there's a path from top-left to bottom-right cell.
7. Find if a cycle exists in a grid (4-directional moves).
8. Number of ways to reach bottom-right from top-left (DFS style).
9. Check if maze with walls and open paths has an escape path.
10. Print the path using BFS from source to destination in a grid.

🔒 *Obstacles and Movement Constraints*

11. Traverse a grid avoiding blocked cells (value `-1` or `#`).
12. Count the shortest distance to all reachable cells from a source.
13. Knight's shortest path on a chessboard from (x1,y1) to (x2,y2).
14. Find if rat can reach cheese with only L or R or D moves.
15. In a grid with teleporters, find shortest path using BFS.
16. Flood fill with diagonal connectivity (8 directions).
17. BFS to simulate fire spreading from multiple sources.
18. Person escaping a burning maze using BFS + simulation.
19. Simulate zombie infection spreading in a grid day-by-day.
20. Count all cells within distance ≤ `k` from a given cell.

Pattern Problems / Games / Connectivity

21. Tic-Tac-Toe winner detection using grid checks.
22. Count all 2x2 blocks with exactly one `#`.
23. Validate if a grid is bipartite using BFS coloring.
24. In a grid of heights, mark all cells reachable from both oceans.
25. In a grid of 1s/0s, find the largest connected region (max area).
26. Find all boundary cells in a connected region of `1`s.
27. Detect "closed islands" (regions of 1s not touching boundary).
28. Simulate Conway's Game of Life for 1 step.
29. Recolor a border of connected region in the same color.
30. Find the maximum number of steps from a start cell to any reachable cell.

Would you like:

- BFS and DFS templates for grid traversal (in C++ or Java)?
- Link to Codeforces / AtCoder / LeetCode 1100–1200 grid problems?
- Python-style pseudocode for faster practice?

Let me know your preferred format or help with any specific problem!

Tools