

Perfect 🍀 — you want **50 Codeforces-style array problem statements** with **constraints** (like $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, etc.) — for **competitive programming practice**. Here's a full curated list divided by **difficulty level** (Easy → Hard), each with **realistic constraints** and **short descriptions** similar to Codeforces problems.

□ EASY LEVEL (1–15)

1. Sum of Elements

Given an array of size n , find the sum of all elements.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $|a_i| \leq 10^9$, $|a_i| \leq 10^9$

2. Maximum Element

Find the maximum element in the array.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

3. Minimum and Maximum Difference

Find the difference between the maximum and minimum elements.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

4. Count Even Numbers

Count how many even numbers exist in the array.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$, $1 \leq a_i \leq 10^9$

5. Reverse the Array

Print the array elements in reverse order.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

6. Sum of Positive Numbers

Compute the sum of all positive numbers in the array.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

7. Unique Elements Count

Count distinct elements in the array.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $0 \leq a_i \leq 10^9$, $0 \leq a_i \leq 10^9$

8. First Repeated Element

Find the first element that appears more than once.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

9. Prefix Sum Query

Given an array and q queries, each query asks for the sum of elements from l to r .

Constraints: $1 \leq n, q \leq 10^5$, $1 \leq n, q \leq 10^5$

10. Array Rotation

Rotate the array right by k steps.

Constraints: $1 \leq n \leq 10^5$, $0 \leq k < n$, $1 \leq n \leq 10^5$, $0 \leq k < n$

11. Check Sorted

Check if the array is sorted in non-decreasing order.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

12. Count Negatives

Count how many negative numbers are in the array.

Constraints: $1 \leq n \leq 10^5$, $-10^9 \leq a_i \leq 10^9$, $1 \leq n \leq 10^5$, $-10^9 \leq a_i \leq 10^9$

13. Sum of Alternate Elements

Find the sum of elements at even indices.

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

14. Product of All Elements (mod M)

Output the product of all array elements modulo $10^9 + 7$.

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$

15. Number of Zeros

Count how many zeros appear in the array.

Constraints: $1 \leq n \leq 10^5$, $0 \leq a_i \leq 10^5$

⚙ MEDIUM LEVEL (16–35)

16. Equal Pair Count

Count how many pairs (i, j) exist such that $a_i = a_j$ and $i < j$.

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

17. Second Largest Element

Find the second largest unique element.

Constraints: $2 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

18. Remove Duplicates

Print the array after removing duplicates while keeping order.

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

19. Frequency Query

Output how many times a given value x appears.

Constraints: $1 \leq n, q \leq 10^5$, $1 \leq a_i \leq 10^5$

20. Balanced Array

Check if the sum of the first half equals the sum of the second half.

Constraints: $2 \leq n \leq 10^5$, n is even, $1 \leq a_i \leq 10^5$

21. Majority Element

Find an element that appears more than $n/2$ times (if exists).

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

22. K-th Largest Element

Find the k -th largest number in the array.

Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$, $1 \leq a_i \leq 10^5$

23. Missing Number

Given integers from 1 to n with one missing, find the missing one.

Constraints: $2 \leq n \leq 10^6$, $1 \leq a_i \leq 10^6$

24. Pair with Given Sum

Check if any two numbers sum to x .

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^9$, $1 \leq x \leq 10^9$

25. Rearrange by Parity

Move all even numbers to the front, odd to the back.

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

26. Subarray Sum Equals K

Count the number of subarrays with sum equal to k .

Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

27. **Longest Increasing Subarray**
Find the length of the longest increasing contiguous subarray.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 28. **Difference Array Operations**
After q range increment operations, output final array.
Constraints: $1 \leq n, q \leq 10^5$, $1 \leq a_i \leq 10^5$
 29. **Sort by Frequency**
Sort elements by descending frequency.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 30. **Minimum Number of Moves to Equal Array**
You can increment/decrement an element by 1. Find min moves to make all equal.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 31. **Cyclic Shift Equality**
Check if one array can be rotated to become another.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 32. **Equalize Array**
Delete minimum elements so that all remaining are equal.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 33. **Remove Adjacent Duplicates**
Continuously remove adjacent duplicates until none remain.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 34. **Smallest Missing Positive**
Find the smallest positive integer missing from the array.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
 35. **Longest Subarray with Equal 0s and 1s**
Given an array of 0s and 1s, find the longest subarray with equal 0s and 1s.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
-

HARD LEVEL (36–50)

36. **Maximum Subarray Sum (Kadane's Algorithm)**
Find the maximum possible subarray sum.
Constraints: $1 \leq n \leq 10^6$, $-10^9 \leq a_i \leq 10^9$
37. **Subarray with Given XOR**
Count subarrays with XOR equal to k .
Constraints: $1 \leq n \leq 10^5$, $0 \leq a_i \leq 10^9$
38. **Number of Inversions**
Count pairs (i, j) where $i < j$ and $a_i > a_j$.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
39. **Find Peak Element**
Find an index i where $a_i > a_{i-1}$ and $a_i > a_{i+1}$.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
40. **Subarray Minimums Sum**
Find sum of minimums of all subarrays.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$

41. **Next Greater Element**
For each element, find the next greater element to its right.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
42. **Trapping Rain Water**
Find how much water can be trapped between bars.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
43. **Maximum Product Subarray**
Find maximum product of a contiguous subarray.
Constraints: $1 \leq n \leq 10^5$, $-10^5 \leq a_i \leq 10^5$
44. **Minimum Swaps to Sort**
Find the minimum number of swaps to sort the array.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
45. **Merge Intervals in Array**
Given intervals $[l, r]$, merge overlapping ones.
Constraints: $1 \leq n \leq 10^5$, $1 \leq l \leq r \leq 10^5$
46. **Sliding Window Maximum**
For window size k , find max in each window.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$, $1 \leq a_i \leq 10^5$
47. **Median of Two Sorted Arrays**
Find the median of two sorted arrays.
Constraints: $1 \leq n, m \leq 10^5$, $1 \leq a_i, b_i \leq 10^5$
48. **Longest Subarray Sum $\leq K$**
Find the longest subarray with sum $\leq K$.
Constraints: $1 \leq n \leq 10^5$, $-10^5 \leq a_i \leq 10^5$, $-10^5 \leq K \leq 10^5$
49. **Split Array into Equal Sum Parts**
Find number of ways to split array into 3 contiguous parts with equal sum.
Constraints: $1 \leq n \leq 10^5$, $1 \leq a_i \leq 10^5$
50. **Maximize Sum after K Negations**
Given k operations, in each you can flip the sign of one element; maximize final sum.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$, $-10^5 \leq a_i \leq 10^5$

Would you like me to include **sample input/output + explanation** for each of these 50 problems (like Codeforces format)?
It'll make them fully ready for coding practice.

You said:

queue

ChatGPT said:

Nice ✨ — you want **50 Codeforces-style problem statements on Queues** (with **constraints**, like real contests).
Here's a full competitive-programming-ready set — from **easy** → **hard**, all based on queue logic, simulation, or advanced applications (monotonic queue, BFS, etc.).

□ EASY LEVEL (1–15)

1. Simple Queue Simulation

Given n enqueue and dequeue operations, print the final contents of the queue.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

2. Front and Back of Queue

Process q queries of 3 types:

- 1 $x \rightarrow$ push x
- 2 \rightarrow pop front
- 3 \rightarrow print front element

Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$

3. Empty or Not

Given q operations, after each one, print “YES” if the queue is empty else “NO”.

Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$

4. Reverse a Queue

Given n integers, reverse them using a queue.

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

5. Queue Size Tracker

Process enqueue/dequeue operations and print current queue size after each operation.

Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$

6. Last Element in Queue

After a series of enqueue/dequeue operations, print the element at the back (or -1 if empty).

Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$

7. Customer Queue at Bank

Each customer takes t_i seconds to serve. Print total waiting time for all customers.

Constraints: $1 \leq n \leq 10^5$, $1 \leq t_i \leq 10^4$, $1 \leq t_i \leq 10^4$, $1 \leq t_i \leq 10^4$

8. Circular Queue Implementation

Simulate a circular queue of capacity k . Output “FULL” or “EMPTY” after each operation.

Constraints: $1 \leq q \leq 10^5$, $1 \leq k \leq 10^5$, $1 \leq k \leq 10^5$, $1 \leq k \leq 10^5$

9. Queue Sum Queries

Support operations: enqueue(x), dequeue(), and print sum of all elements.

Constraints: $1 \leq q \leq 10^5$, $|x| \leq 10^9$, $1 \leq q \leq 10^5$, $|x| \leq 10^9$

10. Check Front and Rear Equal

Check if front and rear elements are equal at any point.

Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$

11. Multiple Queues

Maintain k queues; process enqueue/dequeue operations for specific queues.

Constraints: $1 \leq k, q \leq 10^5$, $1 \leq k, q \leq 10^5$

12. Queue of Strings

Implement queue operations for strings instead of integers.

Constraints: $1 \leq q \leq 10^5$, $|s| \leq 20$, $1 \leq q \leq 10^5$, $|s| \leq 20$

13. Sum of First and Last Element

Print sum of front and back after every operation.

Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$

14. Queue Rotation

Rotate queue k times to the right.

Constraints: $1 \leq n \leq 10^5$, $0 \leq k < n$, $1 \leq n \leq 10^5$, $0 \leq k < n$

15. Copy Queue

Copy contents of one queue into another and print both.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

⚙ MEDIUM LEVEL (16–35)

16. Queue Using Two Stacks

Implement a queue using two stacks (efficiently).

Constraints: $1 \leq q \leq 10^5$ $1 \leq q \leq 10^5$

17. Print Queue Elements in Reverse without Extra Queue

Use recursion to reverse and print queue.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

18. People Waiting for Ticket

Each person has a priority (smaller number = higher priority). Print order of service.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

19. Queue Reordering

Given a queue, bring all even numbers to the front while maintaining order.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

20. Printer Queue Simulation

Given jobs with priorities, print the order in which jobs complete.

Constraints: $1 \leq n \leq 10^5, 1 \leq \text{priority} \leq 10^9$ $1 \leq n \leq 10^5, 1 \leq \text{priority} \leq 10^9$

21. Time to Get Ticket

A person at position kkk in queue wants a ticket. Each person buys one per round.

Find how long it takes for person kkk.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

22. Queue Balancing

Two queues contain integers. You can move front element from one to another.

Find min moves to make their sums equal.

Constraints: $1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$ $1 \leq n \leq 10^5, 1 \leq a_i \leq 10^9$

23. Queue Rotation to Target

Given a queue and a target number, find how many rotations to bring target to front.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

24. Sliding Window Average

For window size kkk, print average of elements in each window using a queue.

Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$ $1 \leq n \leq 10^5, 1 \leq k \leq n$

25. Pair Sum in Queue

Check if there exists two elements in queue that sum to xxx.

Constraints: $1 \leq n \leq 10^5$ $1 \leq n \leq 10^5$

26. First Non-Repeating Character in Stream

For each character input, print first non-repeating character so far.

Constraints: $1 \leq |s| \leq 10^5$ $1 \leq |s| \leq 10^5$

27. Interleave Two Halves of Queue

Given a queue of even size, rearrange elements by interleaving first and second halves.

Constraints: $2 \leq n \leq 10^5$ $2 \leq n \leq 10^5$

28. **Reverse First K Elements of Queue**
Reverse the first k elements, leave rest unchanged.
Constraints: $1 \leq k \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq k \leq n \leq 10^5$
29. **Gas Station Queue**
Each car has fuel and cost to move to next. Find if possible to complete circle.
Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$
30. **Circular Tour**
Similar to Gas Station — find starting point index to complete loop.
Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$
31. **Queue Stability Check**
Given two queues, check if they are rotations of each other.
Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$
32. **Merge Two Queues Alternately**
Merge two queues by taking elements alternately.
Constraints: $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$
33. **Queue of Operations**
Each operation has a delay time. Print order of completion.
Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$
34. **Team Formation using Queues**
Students enter queues based on their teams. Simulate team-based enqueue.
Constraints: $1 \leq n, q \leq 10^5$, $1 \leq n, q \leq 10^5$, $1 \leq n, q \leq 10^5$
35. **Queue to Stack Conversion**
Implement a stack using two queues.
Constraints: $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$, $1 \leq q \leq 10^5$
-

🔥 HARD LEVEL (36–50)

36. **Maximum in Sliding Window (Monotonic Queue)**
Find maximum in every subarray of size k .
Constraints: $1 \leq n \leq 10^6$, $1 \leq k \leq n$, $1 \leq n \leq 10^6$, $1 \leq k \leq n$
37. **Sum of Minimums of Subarrays (Monotonic Queue)**
Find sum of minimum elements in all subarrays.
Constraints: $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$, $1 \leq n \leq 10^5$
38. **Queue-Based BFS Traversal**
Given a graph, perform BFS and print visit order.
Constraints: $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$
39. **Shortest Path in Grid (BFS)**
Given a grid of 0s (free) and 1s (blocked), find shortest path using queue BFS.
Constraints: $1 \leq n, m \leq 10^3$, $1 \leq n, m \leq 10^3$, $1 \leq n, m \leq 10^3$
40. **Rotten Oranges**
Given a grid of oranges, compute minimum time for all to rot using BFS.
Constraints: $1 \leq n, m \leq 10^3$, $1 \leq n, m \leq 10^3$, $1 \leq n, m \leq 10^3$
41. **Course Schedule (Topological Sort with Queue)**
Determine if all courses can be finished given dependencies.
Constraints: $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$, $1 \leq n, m \leq 10^5$

42. **Knight Moves on Chessboard**
Find minimum moves for knight to reach target using queue-based BFS.
Constraints: $1 \leq n \leq 5001 \ \& \ n \leq 5001 \leq n \leq 500$
43. **Shortest Path in Binary Maze**
Find shortest path from source to destination using BFS queue.
Constraints: $1 \leq n, m \leq 1031 \ \& \ n, m \leq 10^31 \leq n, m \leq 103$
44. **Queue Reconstruction by Height**
Given people with (height, count), reconstruct queue order.
Constraints: $1 \leq n \leq 1051 \ \& \ n \leq 10^51 \leq n \leq 105$
45. **Truck Tour (Queue Simulation)**
Each station gives petrol and has a distance. Find first station to complete circle.
Constraints: $1 \leq n \leq 1051 \ \& \ n \leq 10^51 \leq n \leq 105$
46. **Snake and Ladder Game (BFS Queue)**
Find minimum dice throws to reach end.
Constraints: $1 \leq N \leq 1041 \ \& \ N \leq 10^41 \leq N \leq 104$
47. **Minimum Operations to Convert A to B**
Use queue BFS where operations are multiply by 2 or subtract 1.
Constraints: $1 \leq a, b \leq 1061 \ \& \ a, b \leq 10^61 \leq a, b \leq 106$
48. **Queue Using Linked List (Custom Implementation)**
Implement queue without STL, using linked list nodes.
Constraints: $1 \leq q \leq 1051 \ \& \ q \leq 10^51 \leq q \leq 105$
49. **Maximum Distance from Land (Multi-source BFS)**
In a grid of 0s and 1s, find max distance from any 0 to nearest 1 using queue.
Constraints: $1 \leq n, m \leq 10001 \ \& \ n, m \leq 10001 \leq n, m \leq 1000$
50. **Circular Queue Game**
N people stand in a circle. Every k-th person is removed (Josephus problem).
Constraints: $1 \leq n \leq 106, 1 \leq k \leq n1 \ \& \ n \leq 10^6, 1 \leq k \leq n1 \leq n \leq 106, 1 \leq k \leq n$

Would you like me to make **input/output examples + explanation** for these 50 queue problems (in Codeforces-style format)?
It'll make them directly usable for practice or mock contests.

You said:

matrix

ChatGPT said:

Excellent 🍀 — you want **50 Codeforces-style matrix problem statements** with **constraints** (like real contests).

These cover **basic matrix operations** → **submatrix queries** → **graph/grid BFS** → **combinatorics on matrices** — all sorted by **difficulty**.

❑ **EASY LEVEL (1–15)**

1. **Matrix Sum**
Given a matrix of size $n \times m$, print the sum of all its elements.
Constraints: $1 \leq n, m \leq 1000, |a_{ij}| \leq 10^9, 1 \leq i \leq n, 1 \leq j \leq m$
 2. **Row Sum and Column Sum**
For each row and column, print their sums.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 3. **Transpose of Matrix**
Print the transpose of an $n \times m$ matrix.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 4. **Check Symmetric Matrix**
Check if an $n \times n$ matrix is symmetric ($a[i][j] = a[j][i]$).
Constraints: $1 \leq n \leq 1000, 1 \leq i, j \leq n$
 5. **Diagonal Sum**
Print the sum of the primary and secondary diagonals of a square matrix.
Constraints: $1 \leq n \leq 1000, 1 \leq i, j \leq n$
 6. **Identity Matrix Check**
Check whether the given matrix is an identity matrix.
Constraints: $1 \leq n \leq 1000, 1 \leq i, j \leq n$
 7. **Upper and Lower Triangular Matrix Check**
Check if a square matrix is upper or lower triangular.
Constraints: $1 \leq n \leq 1000, 1 \leq i, j \leq n$
 8. **Matrix Multiplication**
Multiply two matrices A ($n \times m$) and B ($m \times p$).
Constraints: $1 \leq n, m, p \leq 200, 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq p$
 9. **Matrix Rotation (90° Clockwise)**
Rotate a square matrix 90° clockwise.
Constraints: $1 \leq n \leq 1000, 1 \leq i, j \leq n$
 10. **Find Maximum Element in Matrix**
Output the largest element in the matrix.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 11. **Replace Negative with Zero**
Replace all negative elements with 0.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 12. **Check Sparse Matrix**
A matrix is sparse if more than half its elements are zero — print “YES” or “NO”.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 13. **Search Element in Matrix**
Given an $n \times m$ sorted matrix, search for element x .
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 14. **Count Odd and Even Elements**
Count how many even and odd numbers exist in the matrix.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
 15. **Matrix Border Sum**
Find the sum of elements on the boundary of the matrix.
Constraints: $1 \leq n, m \leq 1000, 1 \leq i \leq n, 1 \leq j \leq m$
-

⚙ MEDIUM LEVEL (16–35)

16. Add and Subtract Two Matrices

Compute $C=A+BC = A + BC=A+B$ and $D=A-BD = A - BD=A-B$.

Constraints: $1 \leq n, m \leq 5001 \setminus \leq n, m \setminus \leq 5001 \leq n, m \leq 500$

17. Row with Maximum 1s

Each row is sorted (0s followed by 1s). Find the row with maximum 1s.

Constraints: $1 \leq n, m \leq 1031 \setminus \leq n, m \setminus \leq 10^3 1 \leq n, m \leq 103$

18. Column with Maximum Sum

Find the column that has the maximum sum.

Constraints: $1 \leq n, m \leq 1031 \setminus \leq n, m \setminus \leq 10^3 1 \leq n, m \leq 103$

19. Wave Print of Matrix

Print elements column-wise in a wave pattern.

Constraints: $1 \leq n, m \leq 1031 \setminus \leq n, m \setminus \leq 10^3 1 \leq n, m \leq 103$

20. Spiral Order Print

Print the matrix elements in spiral order.

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

21. Matrix Rotation by K (90° each time)

Rotate matrix kkk times by 90° clockwise.

Constraints: $1 \leq n \leq 1000, 1 \leq k \leq 41 \setminus \leq n \setminus \leq 1000, 1 \setminus \leq k \setminus \leq 41 \leq n \leq 1000, 1 \leq k \leq 4$

22. Matrix Reflection

Reflect matrix horizontally or vertically.

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

23. Sum of Each Submatrix (Prefix Sum)

Given queries $(x1, y1, x2, y2)(x1, y1, x2, y2)(x1, y1, x2, y2)$, find the sum of that submatrix.

Constraints: $1 \leq n, m \leq 103, 1 \leq q \leq 1051 \setminus \leq n, m \setminus \leq 10^3, 1 \setminus \leq q \setminus \leq 10^5 1 \leq n, m \leq 103, 1 \leq q \leq 105$

24. Matrix Binary Search

Matrix is sorted row-wise and column-wise. Search for xxx .

Constraints: $1 \leq n, m \leq 1031 \setminus \leq n, m \setminus \leq 10^3 1 \leq n, m \leq 103$

25. Zero Matrix Transformation

If any element is 0, set its entire row and column to 0.

Constraints: $1 \leq n, m \leq 1031 \setminus \leq n, m \setminus \leq 10^3 1 \leq n, m \leq 103$

26. Count Special Elements

An element is special if it's max in its row but min in its column. Count them.

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

27. Sum of Diagonal and Boundary Elements

Output total of both diagonals and boundaries (avoid double-counting corners).

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

28. Matrix Rearrangement by Row Sum

Sort rows of matrix by their row sum.

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

29. Replace Each Element by Sum of Neighbors

Replace each element with sum of its 8 neighbors.

Constraints: $1 \leq n, m \leq 5001 \setminus \leq n, m \setminus \leq 5001 \leq n, m \leq 500$

30. Matrix Row Swap

Given two row indices i, j, j, i , swap them.

Constraints: $1 \leq n, m \leq 10001 \setminus \leq n, m \setminus \leq 10001 \leq n, m \leq 1000$

31. Matrix Column Swap

Given two column indices i, j , swap them.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

32. Sum of All Submatrices

Find sum of all possible submatrices.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

33. Count Islands (Grid of 0s and 1s)

Count connected groups of 1s (using 8-direction DFS/BFS).

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

34. Matrix Path Sum

Find minimum path sum from top-left to bottom-right.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

35. Matrix Prefix XOR Queries

Given matrix of integers, answer XOR of submatrix queries.

Constraints: $1 \leq n, m \leq 500, 1 \leq q \leq 1051 \leq n, m \leq 500, 1 \leq q \leq 10^5$

HARD LEVEL (36–50)

36. Rotate Matrix Layer by Layer (In-place)

Rotate matrix rings without extra space.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

37. Largest Square of 1s (DP)

Given binary matrix, find area of largest square submatrix of 1s.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

38. Maximum Rectangle of 1s

Find area of largest rectangle of 1s in a binary matrix.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

39. Matrix Chain Multiplication (DP)

Given dimensions, find minimum number of multiplications.

Constraints: $1 \leq n \leq 5001 \leq n \leq 500$

40. Flood Fill Algorithm

Given matrix and starting point, fill connected region with new color.

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

41. Shortest Path in Binary Matrix (BFS)

Find shortest path from top-left to bottom-right avoiding blocked cells.

Constraints: $1 \leq n, m \leq 5001 \leq n, m \leq 500$

42. Rotten Oranges Problem

Find time required for all oranges to rot (multi-source BFS).

Constraints: $1 \leq n, m \leq 10001 \leq n, m \leq 1000$

43. Word Search in Matrix

Check if a word exists in the matrix (horizontal/vertical/diagonal).

Constraints: $1 \leq n, m \leq 2001 \leq n, m \leq 200$

44. Spiral Fill of Matrix

Fill numbers from 1 to $n \times n$ in spiral order.

Constraints: $1 \leq n \leq 10001 \leq n \leq 1000$

45. **Zigzag Diagonal Traversal**

Traverse all diagonals in zigzag order.

Constraints: $1 \leq n, m \leq 1000$

46. **Matrix Exponentiation**

Given a square matrix and integer k , compute $A^k \bmod (10^9+7)$.

Constraints: $1 \leq n \leq 100, 1 \leq k \leq 10^9$

47. **Determinant of Matrix (Modular Arithmetic)**

Compute determinant of $n \times n$ matrix modulo 10^9+7 .

Constraints: $1 \leq n \leq 500$

48. **Rank of Matrix (Gaussian Elimination)**

Find rank of given matrix using row-reduction.

Constraints: $1 \leq n, m \leq 500$

49. **Find Path with Maximum Product**

Move only right or down; find path with maximum product.

Constraints: $1 \leq n, m \leq 500$

50. **K-th Smallest Element in Matrix**

Matrix is sorted row-wise and column-wise; find k -th smallest element.

Constraints: $1 \leq n, m \leq 1000, 1 \leq k \leq n \times m$

Would you like me to add **sample input/output** + **explanation** for each of these 50 matrix problems (like Codeforces or LeetCode style)?

It will make them directly usable for daily practice or mock contests.

You said:

stack

ChatGPT said:

Here are **50 competitive-style stack problem statements with realistic Codeforces-like constraints** — covering classic, medium, and advanced stack-based algorithms (Next Greater Element, Histogram, Parentheses Matching, Monotonic Stack, etc.):

□ 50 Stack Problem Statements (Codeforces Style)

◆ Basic Level (1–10)

1. **Simple Stack Operations** — Given a sequence of operations (`push x`, `pop`), simulate the stack and print the remaining elements.
 - Constraints: $1 \leq n \leq 10^5$
2. **Top of Stack** — After a sequence of operations, print the top element or “Empty” if stack is empty.
 - Constraints: $1 \leq n \leq 10^4$

3. **Reverse Stack Elements** — Reverse the contents of a given stack using only stack operations.
 - Constraints: $1 \leq n \leq 10^5$
 4. **Balanced Parentheses** — Check if a string containing `()[]{}` is balanced.
 - Constraints: $1 \leq |s| \leq 10^6$
 5. **Postfix Evaluation** — Evaluate a postfix expression using a stack.
 - Constraints: $1 \leq |\text{expression}| \leq 10^5$
 6. **Infix to Postfix Conversion** — Convert a valid infix expression to postfix using a stack.
 - Constraints: $1 \leq |\text{expression}| \leq 10^5$
 7. **Valid Parentheses with Extra Characters** — Given an expression, ignore non-bracket characters and check if it's balanced.
 - Constraints: $1 \leq |s| \leq 10^6$
 8. **Minimum Stack** — Implement a stack that supports push, pop, top, and retrieving the minimum in $O(1)$.
 - Constraints: $1 \leq n \leq 10^5$
 9. **Stack Permutation** — Determine if a given output sequence can be produced using a stack given an input sequence $1 \dots n$.
 - Constraints: $1 \leq n \leq 10^5$
 10. **Next Greater Element (NGE)** — For each element in array, find the next greater element to its right.
 - Constraints: $1 \leq n \leq 2 \times 10^5, 1 \leq a[i] \leq 10^9$
-

◆ Intermediate Level (11–30)

11. **Next Smaller Element (NSE)** — Find the next smaller element for every element in array.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
12. **Previous Greater Element (PGE)** — For each element, find the nearest greater element to its left.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
13. **Previous Smaller Element (PSE)** — For each element, find nearest smaller element to its left.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
14. **Largest Rectangle in Histogram** — Find the largest rectangular area possible in a histogram.
 - Constraints: $1 \leq n \leq 10^5, 1 \leq h[i] \leq 10^9$
15. **Trapping Rain Water** — Given an array of heights, find total trapped rainwater.
 - Constraints: $1 \leq n \leq 10^5$
16. **Stock Span Problem** — For each day, find how many consecutive days before it have price \leq today's price.
 - Constraints: $1 \leq n \leq 10^5$
17. **Remove Adjacent Duplicates** — Given a string, repeatedly remove adjacent duplicates using a stack.
 - Constraints: $1 \leq |s| \leq 10^6$
18. **Simplify Path** — Simplify a Unix-style file path (`/a/./b/./c/`).
 - Constraints: $1 \leq |\text{path}| \leq 10^5$

19. **Celebrity Problem** — Find if there's a person known by everyone but knows no one (using stack reduction).
 - Constraints: $1 \leq n \leq 10^3$
 20. **Valid String After Removing Stars** — Given a string with *, remove the character before each *.
 21. **Decode String** — Given an encoded string like 3[a2[c]], decode it using stack simulation.
 - Constraints: $1 \leq |s| \leq 10^5$
 22. **Minimum Add to Make Parentheses Valid** — Find minimum insertions to make parentheses balanced.
 - Constraints: $1 \leq |s| \leq 10^6$
 23. **Reverse Substrings Between Parentheses** — Reverse text inside parentheses from innermost to outermost.
 - Constraints: $1 \leq |s| \leq 10^5$
 24. **Remove K Digits** — Given number string and integer k, remove k digits to form smallest possible number.
 - Constraints: $1 \leq |num| \leq 10^5$
 25. **Score of Parentheses** — Compute score of balanced parentheses string where $() = 1$, $(A) = 2 \times A$.
 - Constraints: $1 \leq |s| \leq 10^6$
 26. **Validate Stack Sequences** — Given push/pop sequences, check if they are valid stack operations.
 - Constraints: $1 \leq n \leq 10^5$
 27. **Next Greater Element II (Circular Array)** — Find NGE considering array as circular.
 - Constraints: $1 \leq n \leq 10^5$
 28. **Remove All Adjacent Duplicates in K Group** — Given string, remove adjacent k duplicates repeatedly.
 - Constraints: $1 \leq |s| \leq 10^6$
 29. **Evaluate Expression** — Evaluate arithmetic expression with +, -, *, /, (,).
 - Constraints: $1 \leq |expr| \leq 10^5$
 30. **Basic Calculator II** — Evaluate string with integers and operators +, -, *, / without parentheses.
 - Constraints: $1 \leq |expr| \leq 10^5$
-

◆ Advanced Level (31–50)

31. **Longest Valid Parentheses** — Find length of longest valid parentheses substring.
 - Constraints: $1 \leq |s| \leq 10^6$
32. **132 Pattern** — Check if array contains 132 pattern using stack.
 - Constraints: $1 \leq n \leq 10^5$
33. **Maximal Rectangle in Binary Matrix** — Given a matrix of 0/1, find area of largest rectangle containing only 1s.
 - Constraints: $1 \leq n, m \leq 200$
34. **Asteroid Collision** — Simulate asteroid collisions given directions and sizes using stack.

- Constraints: $1 \leq n \leq 10^5$
 - 35. **Online Stock Span** — Process stock prices online (stream input).
 - Constraints: $1 \leq q \leq 10^5$
 - 36. **Remove Outermost Parentheses** — Remove the outermost parentheses from every primitive substring.
 - Constraints: $1 \leq |s| \leq 10^6$
 - 37. **Monotonic Stack Count** — Count pairs (i, j) where $a[i] > a[j]$ and $i < j$ using monotonic stack logic.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
 - 38. **Min Stack with GetMin in O(1)** — Design a stack with push, pop, getMin in O(1) without using extra space.
 - Constraints: $1 \leq n \leq 10^5$
 - 39. **Next Greater Frequency Element** — For each element, find next element to its right with greater frequency.
 - Constraints: $1 \leq n \leq 10^5, 1 \leq a[i] \leq 10^5$
 - 40. **Sum of Subarray Minimums** — Find sum of minimums of all subarrays.
 - Constraints: $1 \leq n \leq 10^5$
 - 41. **Sum of Subarray Ranges** — Find sum of ranges (max-min) for all subarrays using monotonic stacks.
 - Constraints: $1 \leq n \leq 10^5$
 - 42. **Remove Duplicate Letters** — Remove duplicate letters so that every letter appears once and result is smallest lexicographically.
 - Constraints: $1 \leq |s| \leq 10^5$
 - 43. **Daily Temperatures** — For each day, find how many days until a warmer temperature.
 - Constraints: $1 \leq n \leq 10^5$
 - 44. **Monotone Increasing Digits** — Given n , find largest number $\leq n$ with digits in non-decreasing order.
 - Constraints: $1 \leq n \leq 10^{18}$
 - 45. **Find Nearest Greater Frequency Element (Left)** — Similar to NGE but considering element frequency.
 - Constraints: $1 \leq n \leq 10^5$
 - 46. **Largest Rectangle of 1s in Matrix (Stack-based)** — Use histogram approach for each row.
 - Constraints: $1 \leq n, m \leq 200$
 - 47. **Next Greater Element Distance** — Find the distance (index difference) to the next greater element for each element.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
 - 48. **Car Fleet Problem** — Given position and speed, find number of car fleets reaching the target.
 - Constraints: $1 \leq n \leq 10^5$
 - 49. **Visible People in Queue** — Count how many people each person can see in a queue using stack.
 - Constraints: $1 \leq n \leq 10^5$
 - 50. **Min Add/Remove to Make Parentheses Valid** — Find min number of parentheses to add/remove for valid string.
 - Constraints: $1 \leq |s| \leq 10^6$
-

Would you like me to next provide **50 stack problems with input/output examples and expected results (like Codeforces test format)?**

You said:

tree

ChatGPT said:

Perfect ♣ — here are **50 Codeforces-style Tree problem statements** with **realistic competitive programming constraints**.
They cover **Tree Traversal, LCA, DFS, BFS, Binary Search Tree, Tree DP, Diameter, Subtree Queries**, etc.

🌲 50 Tree Problem Statements (Codeforces Style)

◆ Basic Level (1–10): Tree Foundations

1. **Tree Traversals** — Given a binary tree, print its preorder, inorder, and postorder traversals.
 - Constraints: $1 \leq n \leq 10^5$
 2. **Height of Tree** — Find the height (max depth) of a binary tree.
 - Constraints: $1 \leq n \leq 10^5$
 3. **Count Leaf Nodes** — Given a tree, count total number of leaf nodes.
 - Constraints: $1 \leq n \leq 10^5$
 4. **Check if Binary Tree is Balanced** — A binary tree is balanced if left and right subtree height differs by ≤ 1 .
 - Constraints: $1 \leq n \leq 10^5$
 5. **Sum of All Nodes** — Given a tree, compute the sum of all node values.
 - Constraints: $1 \leq n \leq 10^5, |val| \leq 10^9$
 6. **Mirror Tree** — Convert a given binary tree into its mirror image.
 - Constraints: $1 \leq n \leq 10^5$
 7. **Level Order Traversal** — Print nodes of tree level by level using a queue.
 - Constraints: $1 \leq n \leq 10^5$
 8. **Check if Two Trees are Identical** — Return YES if both trees have same structure and node values.
 - Constraints: $1 \leq n \leq 10^5$
 9. **Parent Array to Tree** — Given parent array representation of a tree, print its adjacency list.
 - Constraints: $1 \leq n \leq 10^5$
 10. **Find Root of Tree** — Given parent array, find the node that is the root (parent = -1).
 - Constraints: $1 \leq n \leq 10^5$
-

◆ Intermediate Level (11–30): Structural + DFS/BFS + Subtrees

11. **Number of Children** — For every node, count how many children it has.
 - Constraints: $1 \leq n \leq 10^5$
12. **Subtree Size** — For every node, find the number of nodes in its subtree.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
13. **Sum of Subtree Nodes** — For each node, find sum of all nodes in its subtree.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
14. **Diameter of Tree** — Find the length of the longest path between any two nodes.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
15. **Check if Tree is Binary Search Tree (BST)** — Verify if given binary tree is BST.
 - Constraints: $1 \leq n \leq 10^5$
16. **Lowest Common Ancestor (LCA)** — Find LCA of two given nodes in a tree.
 - Constraints: $1 \leq n \leq 2 \times 10^5, 1 \leq q \leq 10^5$
17. **Path Sum in Tree** — Check if there exists a root-to-leaf path whose sum equals s .
 - Constraints: $1 \leq n \leq 10^5$
18. **Convert Tree to Array (Euler Tour)** — Perform an Euler Tour and store visiting order of nodes.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
19. **Depth of Each Node** — Find the depth (distance from root) for all nodes.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
20. **Parent of Each Node** — For all nodes (except root), find its parent.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
21. **Check if Tree is Full** — A full tree has either 0 or 2 children per node.
 - Constraints: $1 \leq n \leq 10^5$
22. **Binary Tree Paths** — Print all paths from root to leaf nodes.
 - Constraints: $1 \leq n \leq 10^5$
23. **Tree Center** — Find the center(s) of an unweighted tree.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
24. **Count of Nodes at Each Depth** — Count how many nodes exist at every level.
 - Constraints: $1 \leq n \leq 10^5$
25. **Check if Tree is Symmetric** — Determine if left and right subtrees are mirror images.
 - Constraints: $1 \leq n \leq 10^5$
26. **Distance Between Two Nodes** — Compute shortest path distance between nodes u and v .
 - Constraints: $1 \leq n \leq 2 \times 10^5$
27. **Tree Edge Count** — Verify that number of edges is $n-1$.
 - Constraints: $1 \leq n \leq 10^5$
28. **K-th Ancestor Query** — For queries (u, k) , find the k -th ancestor of node u .
 - Constraints: $1 \leq n, q \leq 2 \times 10^5$
29. **Tree Leaves in Order** — List all leaf nodes in left-to-right order.
 - Constraints: $1 \leq n \leq 10^5$
30. **Zig-Zag Traversal** — Print nodes level by level but alternate direction each time.
 - Constraints: $1 \leq n \leq 10^5$

◆ Advanced Level (31–50): DP, Heavy-Light, Binary Lifting, Subtree Queries

31. **Binary Lifting Preprocessing** — Preprocess tree for $O(\log n)$ LCA queries.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
32. **Tree Flattening with Euler Tour + Range Updates** — Flatten tree and perform subtree queries using segment tree.
 - Constraints: $1 \leq n, q \leq 2 \times 10^5$
33. **Subtree Sum Queries** — Given value of each node, answer sum queries for subtree rooted at node v .
 - Constraints: $1 \leq n, q \leq 2 \times 10^5$
34. **Path Queries (Sum on Path)** — Given node values, answer sum on path $u-v$.
 - Constraints: $1 \leq n, q \leq 2 \times 10^5$
35. **Tree Diameter Using Two BFS** — Compute diameter using two BFS passes.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
36. **Find All Articulation Points in Tree** — Identify nodes that increase connected components when removed.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
37. **Find All Bridges** — Identify all edges that disconnect the tree if removed.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
38. **Binary Tree to Doubly Linked List** — Convert binary tree to DLL in inorder traversal order.
 - Constraints: $1 \leq n \leq 10^5$
39. **Count Paths with Sum k** — Count all paths in tree that sum to k .
 - Constraints: $1 \leq n \leq 10^5$
40. **Tree Centroid Decomposition** — Decompose the tree into centroids.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
41. **Tree DP – Longest Path from Each Node** — For every node, compute the length of the longest path starting from it.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
42. **Tree Rerooting DP** — Compute maximum distance sum when each node is considered as root.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
43. **Count Nodes at Distance k** — Given node u and integer k , count nodes at distance k from u .
 - Constraints: $1 \leq n \leq 2 \times 10^5$
44. **Find Tree Diameter Path (not just length)** — Output nodes forming the diameter path.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
45. **Weighted Tree Diameter** — Find diameter where each edge has a weight.
 - Constraints: $1 \leq n \leq 2 \times 10^5, 1 \leq w \leq 10^9$
46. **Check for Binary Heap Property** — Verify if given binary tree satisfies min-heap or max-heap property.
 - Constraints: $1 \leq n \leq 10^5$
47. **Count Good Nodes** — Node v is good if its value \geq all ancestors. Count such nodes.
 - Constraints: $1 \leq n \leq 10^5$
48. **Subtree Even Count** — Count how many nodes have an even-sized subtree.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
49. **Find Tree Diameter Using DP on Tree** — Use recursive DP to find diameter.
 - Constraints: $1 \leq n \leq 2 \times 10^5$
50. **Tree Coloring Problem** — Color nodes in such a way that adjacent nodes have different colors (2-coloring check).

- Constraints: $1 \leq n \leq 10^5$

Would you like me to next give **50 tree problems with input/output examples and expected answers (like Codeforces test cases)?**

It'll include sample test formats like:

```
Input
5
1 2
1 3
3 4
3 5
Output
Height = 3
```

?

You said:

Linked list

ChatGPT said:

Excellent ☞ — here are **50 Codeforces-style Linked List problem statements** with **realistic competitive programming constraints**, divided by difficulty:
Covers **singly, doubly, circular linked lists**, and **advanced pointer/algorithm problems** (merge, reverse, detect cycle, etc.)

☞ 50 Linked List Problem Statements (Codeforces Style)

◆ Basic Level (1–10): Core Operations

1. **Create a Linked List** — Construct a singly linked list from given elements and print it.
 - Constraints: $1 \leq n \leq 10^5$
2. **Insert at Beginning** — Insert a new node at the beginning of the linked list.
 - Constraints: $1 \leq n \leq 10^5$
3. **Insert at End** — Insert a new node at the end of the linked list.
 - Constraints: $1 \leq n \leq 10^5$
4. **Insert at Position** — Insert a node at a given 1-based index position.
 - Constraints: $1 \leq n \leq 10^5$
5. **Delete from Beginning** — Delete the first node of a linked list.
 - Constraints: $1 \leq n \leq 10^5$
6. **Delete from End** — Delete the last node of the linked list.

- Constraints: $1 \leq n \leq 10^5$
 - 7. **Delete by Value** — Delete first occurrence of a given value from linked list.
 - Constraints: $1 \leq n \leq 10^5$, $|a[i]| \leq 10^9$
 - 8. **Search Element** — Search for an element in linked list and return its position.
 - Constraints: $1 \leq n \leq 10^5$
 - 9. **Length of Linked List** — Find total number of nodes.
 - Constraints: $1 \leq n \leq 10^5$
 - 10. **Display Linked List in Reverse (Recursive)** — Print elements in reverse order using recursion.
 - Constraints: $1 \leq n \leq 10^5$
-

◆ Intermediate Level (11–30): Two Pointers, Merging, and Reversal

- 11. **Reverse Linked List (Iterative)** — Reverse a singly linked list iteratively.
 - Constraints: $1 \leq n \leq 10^5$
- 12. **Reverse Linked List (Recursive)** — Reverse using recursion.
 - Constraints: $1 \leq n \leq 10^5$
- 13. **Middle Element of Linked List** — Find the middle node using slow/fast pointer.
 - Constraints: $1 \leq n \leq 10^5$
- 14. **Nth Node from End** — Find nth node from end using two-pointer method.
 - Constraints: $1 \leq n \leq 10^5$
- 15. **Remove Duplicates (Sorted List)** — Remove duplicates from a sorted linked list.
 - Constraints: $1 \leq n \leq 10^5$
- 16. **Remove Duplicates (Unsorted List)** — Remove duplicates from unsorted list using hash set.
 - Constraints: $1 \leq n \leq 10^5$
- 17. **Merge Two Sorted Lists** — Merge two sorted linked lists into one sorted list.
 - Constraints: $1 \leq n, m \leq 10^5$
- 18. **Intersection of Two Linked Lists** — Find intersection point where two linked lists meet.
 - Constraints: $1 \leq n, m \leq 10^5$
- 19. **Detect Cycle in Linked List** — Check if the list has a cycle using Floyd's cycle detection.
 - Constraints: $1 \leq n \leq 10^5$
- 20. **Remove Cycle in Linked List** — Detect and remove loop without losing nodes.
 - Constraints: $1 \leq n \leq 10^5$
- 21. **Find Starting Node of Cycle** — If cycle exists, find node where cycle begins.
 - Constraints: $1 \leq n \leq 10^5$
- 22. **Palindrome Linked List** — Check if list reads same forward and backward.
 - Constraints: $1 \leq n \leq 10^5$
- 23. **Rotate Linked List by K Places** — Rotate list to the right by k nodes.
 - Constraints: $1 \leq n \leq 10^5$
- 24. **Swap Two Nodes (by Data)** — Swap two nodes given their values.
 - Constraints: $1 \leq n \leq 10^5$
- 25. **Swap Nodes in Pairs** — Swap every two adjacent nodes.
 - Constraints: $1 \leq n \leq 10^5$
- 26. **Reverse Nodes in K-Groups** — Reverse nodes in groups of size k .

- Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
 - 27. **Sort Linked List (Merge Sort)** — Sort list using merge sort algorithm.
 - Constraints: $1 \leq n \leq 10^5$
 - 28. **Add Two Numbers (Linked Lists)** — Two linked lists represent integers in reverse order. Add them.
 - Constraints: $1 \leq n, m \leq 10^5$
 - 29. **Remove Nth Node from End** — Delete nth node from end of list.
 - Constraints: $1 \leq n \leq 10^5$
 - 30. **Partition List Around Value X** — Rearrange so that nodes $< x$ come before $\geq x$.
 - Constraints: $1 \leq n \leq 10^5$
-

◆ Advanced Level (31–50): Doubly, Circular, and Structural Challenges

- 31. **Implement Doubly Linked List** — Perform insert/delete at both ends.
 - Constraints: $1 \leq n \leq 10^5$
- 32. **Insert at Beginning (Doubly List)** — Insert a node at beginning.
 - Constraints: $1 \leq n \leq 10^5$
- 33. **Insert at End (Doubly List)** — Insert a node at the end.
 - Constraints: $1 \leq n \leq 10^5$
- 34. **Delete Given Node (Doubly List)** — Delete a specific node in $O(1)$.
 - Constraints: $1 \leq n \leq 10^5$
- 35. **Reverse Doubly Linked List** — Reverse using next and prev pointers.
 - Constraints: $1 \leq n \leq 10^5$
- 36. **Convert Array to Doubly Linked List** — Create DLL from given array.
 - Constraints: $1 \leq n \leq 10^5$
- 37. **Implement Circular Linked List** — Insert nodes to form a circular linked list.
 - Constraints: $1 \leq n \leq 10^5$
- 38. **Detect Circular List** — Check if given list is circular.
 - Constraints: $1 \leq n \leq 10^5$
- 39. **Split Circular Linked List** — Split circular list into two halves.
 - Constraints: $1 \leq n \leq 10^5$
- 40. **Josephus Problem (Circular List)** — Eliminate every k-th node until one remains.
 - Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
- 41. **Flatten Multilevel Linked List** — Flatten list where each node has a child pointer.
 - Constraints: $1 \leq n \leq 10^5$
- 42. **Clone Linked List with Random Pointers** — Deep copy a list with random pointers.
 - Constraints: $1 \leq n \leq 10^5$
- 43. **Intersection of Two Sorted Lists (Value-wise)** — Find intersection list (common elements).
 - Constraints: $1 \leq n, m \leq 10^5$
- 44. **Union of Two Sorted Lists** — Merge two lists and remove duplicates.
 - Constraints: $1 \leq n, m \leq 10^5$
- 45. **Find Merge Point of Two Lists** — Return node where two lists merge.
 - Constraints: $1 \leq n, m \leq 10^5$
- 46. **Segregate Even and Odd Nodes** — Rearrange so that all even-valued nodes come before odd-valued ones.

- Constraints: $1 \leq n \leq 10^5$
- 47. **Intersection Point in Y-Shaped Linked List** — Find node where lists join (classic Y problem).
 - Constraints: $1 \leq n, m \leq 10^5$
- 48. **Check Circular Palindrome List** — Check if circular linked list is palindrome.
 - Constraints: $1 \leq n \leq 10^5$
- 49. **Find Pair Sum Equal to K (Sorted DLL)** — Check if there exists pair with sum = K.
 - Constraints: $1 \leq n \leq 10^5$
- 50. **LRU Cache (Linked List + HashMap)** — Implement LRU cache using doubly linked list.
 - Constraints: $1 \leq n \leq 10^5, 1 \leq \text{capacity} \leq 10^4$

Would you like me to next give **50 Linked List problems with input/output examples (like Codeforces test cases)** — for example:

```
Input
5
1 2 3 4 5
reverse
Output
5 4 3 2 1
```

?

You said:

HashMap

ChatGPT said:

Here are **50 HashMap problem statements** (with Codeforces-style constraints) — ranging from beginner to advanced — designed for **Java or C++** practice 📄

☐ 50 HashMap Problem Statements

Basic Level (1–10)

1. Count the frequency of each element in an array.
Constraints: $1 \leq n \leq 10^5, 1 \leq \text{arr}[i] \leq 10^9$
2. Given a string, count occurrences of each character.
Constraints: $1 \leq |s| \leq 10^5$
3. Check if two strings are anagrams using HashMap.
Constraints: $1 \leq |s|, |t| \leq 10^5$

4. Find the first non-repeating character in a string.
Constraints: $1 \leq |s| \leq 10^5$
 5. Given an array, find if there are two numbers with sum = k.
Constraints: $1 \leq n \leq 10^5$, $-10^9 \leq \text{arr}[i]$, $k \leq 10^9$
 6. Find number of pairs with given difference k.
Constraints: $1 \leq n \leq 10^5$
 7. Given an array, find the element with maximum frequency.
Constraints: $1 \leq n \leq 10^5$
 8. Count distinct elements in each window of size k.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 9. Determine if array contains duplicate elements.
Constraints: $1 \leq n \leq 10^5$
 10. Remove all duplicate elements from an array while preserving order.
Constraints: $1 \leq n \leq 10^5$
-

Intermediate Level (11–30)

11. Count subarrays with sum equal to k.
Constraints: $1 \leq n \leq 10^5$, $-10^9 \leq \text{arr}[i] \leq 10^9$
12. Find length of longest subarray with sum = 0.
Constraints: $1 \leq n \leq 10^5$
13. Find length of longest substring without repeating characters.
Constraints: $1 \leq |s| \leq 10^5$
14. Group all words that are anagrams.
Constraints: $1 \leq n \leq 10^4$, $1 \leq |\text{word}| \leq 100$
15. Find the intersection of two arrays.
Constraints: $1 \leq n, m \leq 10^5$
16. Given a binary array, count subarrays with equal 0s and 1s.
Constraints: $1 \leq n \leq 10^5$
17. Find common elements in three sorted arrays using HashMap.
Constraints: $1 \leq n_1, n_2, n_3 \leq 10^5$
18. Given a string, find all substrings that occur more than once.
Constraints: $1 \leq |s| \leq 10^4$
19. Count number of subarrays having XOR = k.
Constraints: $1 \leq n \leq 10^5$, $0 \leq \text{arr}[i]$, $k \leq 10^9$
20. Find the longest consecutive sequence in an unsorted array.
Constraints: $1 \leq n \leq 10^5$
21. Given an array, find if any two elements occur more than k distance apart.
Constraints: $1 \leq n \leq 10^5$
22. Find frequency of each word in a sentence.
Constraints: $1 \leq |\text{sentence}| \leq 10^5$
23. Find all pairs of elements whose product is divisible by k.
Constraints: $1 \leq n \leq 10^5$
24. Count distinct elements in each prefix of an array.
Constraints: $1 \leq n \leq 10^5$
25. Find subarrays with equal number of even and odd numbers.
Constraints: $1 \leq n \leq 10^5$

26. Find maximum distance between two equal elements.
Constraints: $1 \leq n \leq 10^5$
27. Find elements occurring more than $\lfloor n/3 \rfloor$ times.
Constraints: $1 \leq n \leq 10^5$
28. Given an array, find the smallest subarray with same degree as array.
Constraints: $1 \leq n \leq 10^5$
29. Find all elements that appear exactly twice.
Constraints: $1 \leq n \leq 10^5$
30. Given a pattern and a string, determine if the string follows the pattern.
Constraints: $1 \leq |\text{pattern}|, |s| \leq 10^4$
-

Advanced Level (31–50)

31. Implement LRU Cache using HashMap + LinkedList.
Constraints: $1 \leq \text{capacity} \leq 10^4, 1 \leq \text{operations} \leq 10^5$
32. Implement LFU Cache using HashMap + frequency map.
Constraints: same as above
33. Find number of subarrays with at most k distinct integers.
Constraints: $1 \leq n \leq 10^5$
34. Find number of subarrays with exactly k distinct integers.
Constraints: $1 \leq n \leq 10^5$
35. Given a list of emails, merge accounts with common emails.
Constraints: $1 \leq n \leq 10^4, 1 \leq |\text{email}| \leq 100$
36. Find all unique triplets that sum to zero.
Constraints: $1 \leq n \leq 10^4$
37. Find all pairs of strings that are rotations of each other.
Constraints: $1 \leq n \leq 10^4$
38. Given two arrays, find pairs (a, b) such that $a + b = k$.
Constraints: $1 \leq n, m \leq 10^5$
39. Find the number of substrings with exactly k distinct characters.
Constraints: $1 \leq |s| \leq 10^5$
40. Detect if a subarray with sum divisible by k exists.
Constraints: $1 \leq n \leq 10^5$
41. Find all subarrays with equal number of 1s, 2s, and 3s.
Constraints: $1 \leq n \leq 10^5$
42. Given two strings, check if they are isomorphic.
Constraints: $1 \leq |s|, |t| \leq 10^5$
43. Given an array, find the count of elements smaller than the current element on the right side.
Constraints: $1 \leq n \leq 10^5$
44. Count number of pairs with equal sum.
Constraints: $1 \leq n \leq 10^5$
45. Find the subarray with maximum frequency of repeating elements.
Constraints: $1 \leq n \leq 10^5$
46. Design a system to count unique users visiting a website per day.
Constraints: $1 \leq n \leq 10^5$
47. Implement a phone directory using HashMap.
Constraints: up to 10^5 entries

48. Given list of students and marks, find top k scorers.
Constraints: $1 \leq n \leq 10^5$
49. Given a sequence of words, find the most common bigram (pair of consecutive words).
Constraints: $1 \leq |\text{text}| \leq 10^5$
50. Implement a word frequency counter that supports updates (insert/delete words dynamically).
Constraints: $1 \leq q \leq 10^5$
-

Would you like me to include **sample input/output examples for 10 of these HashMap problems** (like in Codeforces format)?
That will help you understand their competitive structure.

You said:

BST

ChatGPT said:

Here are **50 Binary Search Tree (BST) problem statements** with **Codeforces-style constraints**, ranging from basic to advanced — perfect for Java or C++ practice 📖

♣ 50 BST Problem Statements

□ Basic Level (1–10)

1. **Insert a node into a Binary Search Tree.**
Constraints: $1 \leq n \leq 10^4$, $1 \leq \text{node value} \leq 10^9$
2. **Search for a value in a BST.**
Constraints: $1 \leq n \leq 10^4$
3. **Find the minimum and maximum value in a BST.**
Constraints: $1 \leq n \leq 10^5$
4. **Find the height of a BST.**
Constraints: $1 \leq n \leq 10^5$
5. **Perform inorder, preorder, and postorder traversal of a BST.**
Constraints: $1 \leq n \leq 10^5$
6. **Count total number of nodes in a BST.**
Constraints: $1 \leq n \leq 10^5$
7. **Find the sum of all nodes in a BST.**
Constraints: $1 \leq n \leq 10^5$
8. **Check if a binary tree is a valid BST.**
Constraints: $1 \leq n \leq 10^5$

9. Find the kth smallest element in a BST.

Constraints: $1 \leq n \leq 10^5$

10. Find the kth largest element in a BST.

Constraints: $1 \leq n \leq 10^5$

□ Intermediate Level (11–30)

11. Delete a node from a BST.

Constraints: $1 \leq n \leq 10^5$

12. Find the lowest common ancestor (LCA) of two nodes in a BST.

Constraints: $1 \leq n \leq 10^5$

13. Check if two BSTs are identical.

Constraints: $1 \leq n \leq 10^5$

14. Convert a BST to a sorted doubly linked list.

Constraints: $1 \leq n \leq 10^5$

15. Convert a sorted array to a balanced BST.

Constraints: $1 \leq n \leq 10^5$

16. Find the floor and ceil of a given key in a BST.

Constraints: $1 \leq n \leq 10^5$

17. Find the inorder successor of a given node.

Constraints: $1 \leq n \leq 10^5$

18. Find the inorder predecessor of a given node.

Constraints: $1 \leq n \leq 10^5$

19. Check if a BST is height-balanced.

Constraints: $1 \leq n \leq 10^5$

20. Print all leaf nodes of a BST.

Constraints: $1 \leq n \leq 10^5$

21. Find distance between two nodes in a BST.

Constraints: $1 \leq n \leq 10^5$

22. Find the diameter (longest path) of a BST.

Constraints: $1 \leq n \leq 10^5$

23. Count the number of nodes in a given range [L, R].

Constraints: $1 \leq n \leq 10^5$

24. Find sum of all nodes within range [L, R].

Constraints: $1 \leq n \leq 10^5$

25. Find the closest value to a given target in BST.

Constraints: $1 \leq n \leq 10^5$

26. Check if a BST contains dead end (no further insert possible).

Constraints: $1 \leq n \leq 10^5$

27. Print BST elements in descending order.

Constraints: $1 \leq n \leq 10^5$

28. Count pairs of nodes from two BSTs whose sum equals X.

Constraints: $1 \leq n_1, n_2 \leq 10^5$

29. Find median of BST.

Constraints: $1 \leq n \leq 10^5$

30. Convert BST to Min Heap.

Constraints: $1 \leq n \leq 10^5$

● Advanced Level (31–50)

31. **Construct BST from preorder traversal.**
Constraints: $1 \leq n \leq 10^5$
32. **Construct BST from postorder traversal.**
Constraints: $1 \leq n \leq 10^5$
33. **Construct BST from level order traversal.**
Constraints: $1 \leq n \leq 10^5$
34. **Serialize and deserialize a BST.**
Constraints: $1 \leq n \leq 10^5$
35. **Find the largest BST in a binary tree.**
Constraints: $1 \leq n \leq 10^5$
36. **Merge two balanced BSTs into one balanced BST.**
Constraints: $1 \leq n_1, n_2 \leq 10^5$
37. **Find the mode(s) (most frequent element) in a BST.**
Constraints: $1 \leq n \leq 10^5$
38. **Find pairs of nodes whose sum is equal to target K.**
Constraints: $1 \leq n \leq 10^5$
39. **Flatten a BST into a sorted linked list.**
Constraints: $1 \leq n \leq 10^5$
40. **Recover BST where two nodes are swapped.**
Constraints: $1 \leq n \leq 10^5$
41. **Convert BST to Greater Sum Tree (GST).**
Constraints: $1 \leq n \leq 10^5$
42. **Find minimum absolute difference between any two nodes.**
Constraints: $1 \leq n \leq 10^5$
43. **Find the path from root to a given node in BST.**
Constraints: $1 \leq n \leq 10^5$
44. **Find all root-to-leaf paths in BST.**
Constraints: $1 \leq n \leq 10^5$
45. **Check if BST is symmetric (mirror image).**
Constraints: $1 \leq n \leq 10^5$
46. **Print boundary traversal of BST.**
Constraints: $1 \leq n \leq 10^5$
47. **Count number of unique BSTs possible with n distinct keys.**
Constraints: $1 \leq n \leq 10^3$
48. **Generate all structurally unique BSTs with values 1 to n.**
Constraints: $1 \leq n \leq 10$
49. **Implement an iterator for BST (BSTIterator class).**
Constraints: $1 \leq n \leq 10^5$
50. **Find maximum sum path in BST (from any node to any node).**
Constraints: $1 \leq n \leq 10^5$

Would you like me to provide **10 sample BST problems with input/output examples** (like Codeforces format)?

That will make them ready for implementation and practice.

You said:

heap

ChatGPT said:

Here's a list of **50 Heap / Priority Queue problem statements** with **Codeforces-style constraints**, ranging from basic to advanced. Perfect for Java, C++, or Python practice. 🏠

☐ 50 Heap / Priority Queue Problems

☐ Basic Level (1–10)

1. **Implement Min Heap** — Insert elements and extract min.
Constraints: $1 \leq n \leq 10^5$
 2. **Implement Max Heap** — Insert elements and extract max.
Constraints: $1 \leq n \leq 10^5$
 3. **Find k smallest elements in an array** using a heap.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
 4. **Find k largest elements in an array** using a heap.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
 5. **Heapify an array** — Convert a given array into a min heap.
Constraints: $1 \leq n \leq 10^5$
 6. **Heapify an array** — Convert a given array into a max heap.
Constraints: $1 \leq n \leq 10^5$
 7. **Find the median in a stream of integers** using two heaps.
Constraints: $1 \leq n \leq 10^5$
 8. **Kth largest element in a stream** — Maintain heap to extract kth largest dynamically.
Constraints: $1 \leq n \leq 10^5$
 9. **Sort a nearly sorted array** (every element at most k positions away).
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
 10. **Priority queue operations** — Implement insert, remove, and peek efficiently.
Constraints: $1 \leq n \leq 10^5$
-

☐ Intermediate Level (11–30)

11. **Merge k sorted arrays** using a min heap.
Constraints: $1 \leq k \leq 10^3$, total elements $\leq 10^5$
12. **Top K frequent elements** from an array using a heap.
Constraints: $1 \leq n \leq 10^5$
13. **Find k closest numbers to X in a sorted array** using heap.
Constraints: $1 \leq n \leq 10^5$
14. **Sliding window maximum** — Find maximum in every window of size k.
Constraints: $1 \leq n \leq 10^5$

15. **Sliding window minimum** — Find minimum in every window of size k .
Constraints: $1 \leq n \leq 10^5$
 16. **Find k th smallest/largest in BST** using min/max heap.
Constraints: $1 \leq n \leq 10^5$
 17. **Reorganize string with no adjacent duplicates** using max heap.
Constraints: $1 \leq |s| \leq 10^5$
 18. **Connect ropes with minimum cost** using min heap.
Constraints: $1 \leq n \leq 10^5$
 19. **Find k smallest pairs from two sorted arrays** using min heap.
Constraints: $1 \leq n, m \leq 10^5, 1 \leq k \leq n \times m$
 20. **K th smallest/largest element in matrix** where rows & columns sorted.
Constraints: $1 \leq n, m \leq 10^3$
 21. **Find k pairs with largest sums** from two arrays using max heap.
Constraints: $1 \leq n, m \leq 10^5$
 22. **Sort nearly sorted sequence (priority queue)**
Constraints: $1 \leq n \leq 10^5, k \leq n$
 23. **Find median of two sorted arrays** using heaps.
Constraints: $1 \leq n, m \leq 10^5$
 24. **Task scheduler (rearrange tasks with cooling period)** using heap.
Constraints: $1 \leq |\text{tasks}| \leq 10^5$
 25. **Rearrange characters in string** to avoid k distance repetition.
Constraints: $1 \leq |s| \leq 10^5$
 26. **Top k largest sums of subarrays** using heap.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq 10^5$
 27. **Find k smallest elements in unsorted stream** using max heap.
Constraints: $1 \leq n \leq 10^5$
 28. **Find k largest elements in unsorted stream** using min heap.
Constraints: $1 \leq n \leq 10^5$
 29. **Task scheduling with priority queue** — execute tasks in priority order.
Constraints: $1 \leq n \leq 10^5$
 30. **Merge k sorted linked lists** using heap.
Constraints: $1 \leq k \leq 10^3, \text{total nodes} \leq 10^5$
-

● Advanced Level (31–50)

31. **Find k closest points to origin** using min/max heap.
Constraints: $1 \leq n \leq 10^5$
32. **Sliding window median** using two heaps.
Constraints: $1 \leq n \leq 10^5$
33. **Minimum cost to hire k workers** using heap.
Constraints: $1 \leq n \leq 10^5$
34. **Maximum sum of k non-overlapping subarrays** using heap.
Constraints: $1 \leq n \leq 10^5$
35. **Implement Dijkstra's algorithm** using min heap.
Constraints: $1 \leq n \leq 10^5, 1 \leq m \leq 2 \times 10^5$
36. **Find k th largest element in min heap** without changing the heap.
Constraints: $1 \leq n \leq 10^5$

37. **Find kth smallest element in max heap** without changing the heap.
Constraints: $1 \leq n \leq 10^5$
38. **Stream of integers, maintain median** dynamically.
Constraints: $1 \leq n \leq 10^5$
39. **Reorder log files** using heap based on priority/lexicographic order.
Constraints: $1 \leq n \leq 10^5$
40. **Kth smallest number in multiplication table** using heap.
Constraints: $1 \leq m, n \leq 10^3$
41. **Online leader board** using max heap.
Constraints: $1 \leq n \leq 10^5$
42. **Maximize sum after k negations** using min heap.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq 10^5$
43. **Sliding window top k frequent elements** using heap.
Constraints: $1 \leq n \leq 10^5$
44. **Merge intervals with earliest end using heap**
Constraints: $1 \leq n \leq 10^5$
45. **Kth largest rectangle area in histogram** using heap.
Constraints: $1 \leq n \leq 10^5$
46. **Kth closest sum to target in array** using heap.
Constraints: $1 \leq n \leq 10^5$
47. **Find top k scores dynamically** in competition using heap.
Constraints: $1 \leq n \leq 10^5$
48. **Streaming top k elements** using heap efficiently.
Constraints: $1 \leq n \leq 10^5$
49. **Median maintenance in sliding window** using heap.
Constraints: $1 \leq n \leq 10^5$
50. **Heap-based scheduling problem** — minimize total waiting time.
Constraints: $1 \leq n \leq 10^5$

If you want, I can next **make 10 Heap problems with input/output examples** in **Codeforces format**, ready to implement and practice.

Do you want me to do that?

You said:

trie

ChatGPT said:

Here's a **list of 50 Trie / Prefix Tree problem statements** with **Codeforces-style constraints**, ranging from basic to advanced. Perfect for Java, C++, or Python practice. 🌲

🌲 50 Trie Problems

□ Basic Level (1–10)

1. **Implement a Trie** — Insert a word and search for exact matches.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 2. **Check if a word exists in Trie** — Return true/false.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 3. **Delete a word from Trie** — Remove a given word.
Constraints: $1 \leq n \leq 10^5$
 4. **Insert multiple words and print all words** stored in Trie.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 5. **Count words with a given prefix** in a Trie.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 6. **Check if Trie contains any word with given prefix.**
Constraints: $1 \leq n \leq 10^5$
 7. **Find the longest common prefix** of all inserted words.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 8. **Count total number of words** stored in Trie.
Constraints: $1 \leq n \leq 10^5$
 9. **Find number of words ending with a specific suffix** using reverse Trie.
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 10. **List all words starting with a given prefix.**
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
-

□ Intermediate Level (11–30)

11. **Autocomplete suggestions** — Given a prefix, return top k lexicographically smallest words.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10$
12. **Implement dictionary with frequency** — Insert words with frequency, search by frequency.
Constraints: $1 \leq n \leq 10^5$
13. **Check if Trie contains palindrome words.**
Constraints: $1 \leq n \leq 10^5$
14. **Count pairs of words where one is prefix of another.**
Constraints: $1 \leq n \leq 10^5$
15. **Find all words that can be formed from a set of characters** using Trie.
Constraints: $1 \leq n \leq 10^5$
16. **Word search in 2D board using Trie** — find all words from dictionary in board.
Constraints: board size $\leq 12 \times 12$, dictionary $\leq 10^4$
17. **Implement spell checker** — Suggest words with edit distance 1.
Constraints: $1 \leq n \leq 10^5$
18. **Longest word formed by other words** — Check if a word can be built from other words in Trie.
Constraints: $1 \leq n \leq 10^5$
19. **Find shortest unique prefix for every word.**
Constraints: $1 \leq n \leq 10^5$
20. **Insert words and count number of words matching exactly or as prefix.**
Constraints: $1 \leq n \leq 10^5$

21. **Check if a string can be segmented into dictionary words** (Word Break problem).
Constraints: $1 \leq |s| \leq 10^5$, dictionary $\leq 10^4$
 22. **Count total number of nodes in Trie.**
Constraints: $1 \leq n \leq 10^5$
 23. **Implement case-insensitive Trie.**
Constraints: $1 \leq n \leq 10^5$
 24. **Check if given word can be formed using letters from multiple words in Trie.**
Constraints: $1 \leq n \leq 10^5$
 25. **Find all anagrams in a dictionary using Trie.**
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 26. **Count words matching a pattern with wildcards ?**
Constraints: $1 \leq n \leq 10^5$, word length ≤ 100
 27. **Find top k frequent words with given prefix** using Trie + heap.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10$
 28. **Trie with numeric keys** — Store and search phone numbers.
Constraints: $1 \leq n \leq 10^5$, number length ≤ 15
 29. **Find all words differing by at most one character from given word.**
Constraints: $1 \leq n \leq 10^5$
 30. **Autocomplete system** — maintain Trie with frequency of searches.
Constraints: $1 \leq n \leq 10^5$
-

● Advanced Level (31–50)

31. **Count total words that are prefixes of other words.**
Constraints: $1 \leq n \leq 10^5$
32. **Find all concatenated words** (words formed by concatenating other words).
Constraints: $1 \leq n \leq 10^5$
33. **Implement Trie with deletion and prefix count in $O(\text{word length})$.**
Constraints: $1 \leq n \leq 10^5$
34. **Dictionary with replacement operations** — Replace words and maintain frequency.
Constraints: $1 \leq n \leq 10^5$
35. **Find maximum length chain of words where each word is prefix of next.**
Constraints: $1 \leq n \leq 10^5$
36. **Find longest palindrome prefix** using Trie + reverse insertions.
Constraints: $1 \leq n \leq 10^5$
37. **Word search with 8-directional movement in grid using Trie.**
Constraints: board size $\leq 12 \times 12$, dictionary $\leq 10^4$
38. **Trie to store IP addresses and find longest common subnet.**
Constraints: $1 \leq n \leq 10^5$
39. **Find all words matching regex pattern** using Trie.
Constraints: $1 \leq n \leq 10^5$
40. **Implement autocomplete with ranking** — Return top k based on frequency and lexicographic order.
Constraints: $1 \leq n \leq 10^5$
41. **Find missing words from paragraph using Trie dictionary.**
Constraints: $1 \leq n \leq 10^5$
42. **Detect words that are prefixes of other words in dictionary.**
Constraints: $1 \leq n \leq 10^5$

43. **Find shortest word not present in dictionary** (minimal string not in Trie).
Constraints: $1 \leq n \leq 10^5$
 44. **Store words with emoji/unicode characters in Trie.**
Constraints: $1 \leq n \leq 10^5$
 45. **Find number of words in Trie with exactly k occurrences of a character.**
Constraints: $1 \leq n \leq 10^5$
 46. **Implement Trie for multi-language dictionary.**
Constraints: $1 \leq n \leq 10^5$
 47. **Find maximum depth of Trie** (longest inserted word).
Constraints: $1 \leq n \leq 10^5$
 48. **Check if a Trie can represent a prefix-free code.**
Constraints: $1 \leq n \leq 10^5$
 49. **Dynamic insert and delete in Trie while maintaining prefix count.**
Constraints: $1 \leq n \leq 10^5$
 50. **Find minimum unique prefix for a new word** before inserting into Trie.
Constraints: $1 \leq n \leq 10^5$
-

If you want, I can **next make 10 Trie problems with input/output examples**, in **Codeforces-style format**, ready to implement.

Do you want me to do that?

You said:

graph

ChatGPT said:

Here's a **list of 50 Graph problem statements** with **Codeforces-style constraints**, ranging from basic to advanced — suitable for Java, C++, or Python practice. 🌐

🌐 50 Graph Problems

□ Basic Level (1–10)

1. **Implement a graph using adjacency list and adjacency matrix.**
Constraints: $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$
2. **Perform BFS traversal** starting from a given node.
Constraints: $1 \leq n \leq 10^5$
3. **Perform DFS traversal** starting from a given node.
Constraints: $1 \leq n \leq 10^5$
4. **Check if a graph is connected** (single component).
Constraints: $1 \leq n \leq 10^5$

5. **Count the number of connected components** in an undirected graph.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
 6. **Check if a graph contains a cycle** (undirected).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
 7. **Check if a graph contains a cycle** (directed).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
 8. **Find shortest path from source to all nodes** in unweighted graph using BFS.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
 9. **Graph representation conversions** — adjacency list \leftrightarrow adjacency matrix.
Constraints: $1 \leq n \leq 10^5$
 10. **Find degree of each vertex** in undirected graph.
Constraints: $1 \leq n \leq 10^5$
-

□ Intermediate Level (11–30)

11. **Topological sort** of a directed acyclic graph (DAG).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
12. **Check if a directed graph is DAG**.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
13. **Detect bridges** (edges whose removal increases number of components).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
14. **Detect articulation points** (vertices whose removal increases components).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
15. **Find strongly connected components (SCC)** in a directed graph (Kosaraju/Tarjan).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
16. **Find number of paths from u to v** in DAG.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
17. **Check if graph is bipartite**.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
18. **Find shortest path in weighted graph** using Dijkstra's algorithm.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 2 \times 10^5$
19. **Find shortest path using Bellman-Ford** (handles negative weights).
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 2 \times 10^5$
20. **Floyd-Warshall algorithm** — all-pairs shortest paths.
Constraints: $1 \leq n \leq 500$
21. **Check if undirected graph is a tree**.
Constraints: $1 \leq n \leq 10^5$, $m = n - 1$
22. **Count number of triangles in undirected graph**.
Constraints: $1 \leq n \leq 10^3$
23. **Find connected components sizes**.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
24. **Graph coloring** — check if graph can be colored with k colors.
Constraints: $1 \leq n \leq 20$
25. **Minimum number of edges to make graph connected**.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$
26. **Check if a graph is Eulerian** — has Eulerian path/circuit.
Constraints: $1 \leq n \leq 10^5$, $0 \leq m \leq 10^5$

27. **Check if a graph is Hamiltonian** (NP-hard, small n).
Constraints: $1 \leq n \leq 20$
 28. **Transpose a directed graph.**
Constraints: $1 \leq n \leq 10^5$
 29. **Graph reachability** — can we reach v from u ?
Constraints: $1 \leq n \leq 10^5$
 30. **Number of paths of length k** in unweighted graph using adjacency matrix exponentiation.
Constraints: $1 \leq n \leq 200, 1 \leq k \leq 10^9$
-

● Advanced Level (31–50)

31. **Kruskal's algorithm** — find Minimum Spanning Tree (MST).
Constraints: $1 \leq n \leq 10^5, 0 \leq m \leq 2 \times 10^5$
32. **Prim's algorithm** — MST using min-heap.
Constraints: $1 \leq n \leq 10^5, 0 \leq m \leq 2 \times 10^5$
33. **Maximum flow (Edmonds-Karp)** — find max flow in network.
Constraints: $1 \leq n \leq 500, 1 \leq m \leq 10^4$
34. **Maximum flow (Dinic's algorithm)** — for larger graphs.
Constraints: $1 \leq n \leq 10^4, 1 \leq m \leq 10^5$
35. **Minimum cut problem** — find min cut in a flow network.
Constraints: $1 \leq n \leq 500, 1 \leq m \leq 10^4$
36. **Check if graph is strongly connected.**
Constraints: $1 \leq n \leq 10^5, 0 \leq m \leq 10^5$
37. **Shortest path with exactly k edges.**
Constraints: $1 \leq n \leq 500, 1 \leq k \leq 100$
38. **All-pairs shortest path in weighted DAG** using DP.
Constraints: $1 \leq n \leq 500$
39. **Graph diameter** — longest shortest path between any two vertices.
Constraints: $1 \leq n \leq 10^5$
40. **Find second shortest path** between two vertices.
Constraints: $1 \leq n \leq 10^5$
41. **Longest path in a DAG** using topological sort.
Constraints: $1 \leq n \leq 10^5$
42. **Check if graph is planar** (small n).
Constraints: $1 \leq n \leq 10$
43. **Traveling Salesman Problem (TSP)** using DP (small n).
Constraints: $1 \leq n \leq 20$
44. **Minimum cost to connect all points** in 2D plane (MST).
Constraints: $1 \leq n \leq 10^4$
45. **Find articulation bridges and points** in weighted graph.
Constraints: $1 \leq n \leq 10^5$
46. **Count number of simple cycles in undirected graph** (small n).
Constraints: $1 \leq n \leq 20$
47. **Dynamic connectivity** — add/remove edges and check connectivity.
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
48. **Weighted graph shortest path with negative cycles** — detect if reachable.
Constraints: $1 \leq n \leq 500, -10^9 \leq \text{weights} \leq 10^9$

49. **Find number of spanning trees** in a graph using Matrix Tree Theorem.
Constraints: $1 \leq n \leq 100$
50. **Graph compression / condensation** — shrink SCCs into single nodes.
Constraints: $1 \leq n \leq 10^5$
-

I can also **create 10 sample Graph problems with Codeforces-style input/output examples** for you, which would make it ready to practice and implement.

Do you want me to do that next?

You said:

Union find

ChatGPT said:

Here's a **list of 50 Union-Find / Disjoint Set Union (DSU) problem statements** with **Codeforces-style constraints**, ranging from basic to advanced — perfect for Java, C++, or Python practice. ☞

☞ 50 Union-Find / DSU Problems

☐ Basic Level (1–10)

1. **Implement basic Union-Find with path compression and union by rank.**
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
2. **Check if two elements are in the same set.**
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
3. **Union two sets containing given elements.**
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
4. **Count number of connected components** after a series of unions.
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
5. **Find size of the set containing a given element.**
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
6. **Check connectivity in a dynamic graph** using DSU.
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
7. **Implement union by size** instead of union by rank.
Constraints: $1 \leq n \leq 10^5$
8. **Count total number of elements in all sets of size $> k$.**
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
9. **Merge multiple sets efficiently** using union-find.
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$

10. **Check if a graph is fully connected** after a series of union operations.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
-

□ **Intermediate Level (11–30)**

11. **Number of connected components after adding edges one by one.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
12. **Count number of redundant connections in a graph** using DSU.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
13. **Detect cycle in an undirected graph** using union-find.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
14. **Implement Kruskal's algorithm** for MST using union-find.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$
15. **Check if adding an edge will create a cycle.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
16. **Find largest connected component** after a series of unions.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
17. **Dynamic connectivity** — Add edges and answer connectivity queries.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
18. **Merge accounts / friends networks** — given pairs of connected elements.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
19. **Find number of sets with size exactly k.**
Constraints: $1 \leq n \leq 10^5$
20. **Check if all nodes in subset are connected.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
21. **Union-find with rollback** — undo last union operations efficiently.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
22. **Find size of largest connected component at each step** of adding edges.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
23. **Number of connected components in 2D grid** (islands problem) using DSU.
Constraints: $1 \leq n, m \leq 1000$
24. **Dynamic connectivity in online graph** — answer queries in real time.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
25. **Check if graph is bipartite using union-find.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
26. **Union-find with additional info per set** — e.g., sum of weights in each set.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
27. **Merging user groups in social network** efficiently.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
28. **Union-find with minimum / maximum element tracking per set.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
29. **Find number of connected components after removing some edges** using DSU on complement graph.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
30. **DSU on tree** — maintain info on subtrees using union-find techniques.
Constraints: $1 \leq n \leq 10^5$
-

● Advanced Level (31–50)

31. **Dynamic connectivity with undo and redo operations.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
32. **Offline dynamic connectivity problem** — batch process queries efficiently.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
33. **Union-find with parity / XOR information** — detect odd/even cycles.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
34. **Check if adding edges will make a graph fully connected at minimum cost.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$
35. **Count number of redundant edges in graph forming MST.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$
36. **Union-find with weighted edges** — track sum of weights per connected component.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
37. **MST with edge replacement queries** — update MST dynamically using DSU.
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 2 \times 10^5$
38. **Offline connectivity with time intervals** — edge exists during certain intervals.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
39. **Count number of distinct sets at each step** after series of union operations.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
40. **Union-find on 2D grid** — count connected components dynamically.
Constraints: $1 \leq n, m \leq 1000$
41. **DSU with rollback for divide and conquer queries.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
42. **Check if graph remains bipartite after each edge addition.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
43. **Dynamic merging of intervals** using union-find.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
44. **DSU for network connectivity with restricted unions.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
45. **Union-find with custom merge function (e.g., max/min/sum).**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
46. **Check if all sets are of equal size after union operations.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
47. **Union-find for counting components with at least k elements.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
48. **Track maximum element in each connected component dynamically.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
49. **Merge friendship circles in social network dynamically** and answer queries.
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
50. **Union-find with rollback and query for sum/max/min in a set.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$

If you want, I can make **10 Union-Find problems with input/output examples** in **Codeforces-style format**, ready to implement and practice.

Do you want me to do that next?

You said:

Two pointer

ChatGPT said:

Here's a **list of 50 Two-Pointer / Sliding Window problem statements** with **Codeforces-style constraints**, ranging from basic to advanced — suitable for Java, C++, or Python practice. ♠

◆ 50 Two-Pointer / Sliding Window Problems

□ Basic Level (1–10)

1. **Find a pair with given sum in a sorted array.**
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
 2. **Find all pairs with sum equal to k in a sorted array.**
Constraints: $1 \leq n \leq 10^5$
 3. **Remove duplicates from sorted array in-place.**
Constraints: $1 \leq n \leq 10^5$
 4. **Move all zeros to the end of array maintaining relative order.**
Constraints: $1 \leq n \leq 10^5$
 5. **Reverse a subarray from index l to r.**
Constraints: $1 \leq n \leq 10^5$
 6. **Merge two sorted arrays into one sorted array in-place.**
Constraints: $1 \leq n, m \leq 10^5$
 7. **Check if array has two elements whose difference is exactly k.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq k \leq 10^9$
 8. **Find maximum sum of any subarray of size k (fixed window size).**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 9. **Count pairs with difference $\leq k$.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq k \leq 10^9$
 10. **Find smallest subarray with sum $\geq S$.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
-

□ Intermediate Level (11–30)

11. **Longest subarray with sum $\leq S$.**
Constraints: $1 \leq n \leq 10^5$
12. **Longest subarray with at most K distinct integers.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^5$
13. **Count subarrays with sum $\leq S$.**
Constraints: $1 \leq n \leq 10^5$

14. **Sort array of 0s, 1s, 2s** (Dutch National Flag problem).
Constraints: $1 \leq n \leq 10^5$
 15. **Find two numbers whose sum is closest to target.**
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
 16. **Find maximum product subarray of size k.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 17. **Container with most water** (LeetCode style).
Constraints: $1 \leq n \leq 10^5$
 18. **Trapping rainwater between bars** using two pointers.
Constraints: $1 \leq n \leq 10^5$
 19. **Check if array contains a continuous subarray sum divisible by k.**
Constraints: $1 \leq n \leq 10^5$
 20. **Partition array into two subsets with equal sum** (use two pointers on sorted array).
Constraints: $1 \leq n \leq 10^5$
 21. **Find all unique triplets with sum 0** (3Sum problem).
Constraints: $1 \leq n \leq 10^5$
 22. **Count number of subarrays with sum exactly K** in non-negative array.
Constraints: $1 \leq n \leq 10^5$
 23. **Find maximum length subarray with sum $\leq K$.**
Constraints: $1 \leq n \leq 10^5$
 24. **Find pairs with sum closest to target in rotated sorted array.**
Constraints: $1 \leq n \leq 10^5$
 25. **Maximize sum of two non-overlapping subarrays of size k.**
Constraints: $1 \leq n \leq 10^5$
 26. **Longest subarray with at most K 0s replaced by 1s.**
Constraints: $1 \leq n \leq 10^5$
 27. **Find minimum window containing all elements of another array.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$
 28. **Subarrays with sum in given range [L, R].**
Constraints: $1 \leq n \leq 10^5$
 29. **Count number of subarrays with exactly K distinct integers.**
Constraints: $1 \leq n \leq 10^5$
 30. **Maximize length of subarray after removing at most k elements.**
Constraints: $1 \leq n \leq 10^5$
-

● Advanced Level (31–50)

31. **Find all quadruplets with sum = target** (4Sum problem).
Constraints: $1 \leq n \leq 10^5$
32. **Maximum sum rectangle in 2D matrix** using sliding window.
Constraints: $1 \leq n, m \leq 500$
33. **Longest substring without repeating characters** using sliding window.
Constraints: $1 \leq |s| \leq 10^5$
34. **Maximum average subarray of size $\geq k$.**
Constraints: $1 \leq n \leq 10^5$
35. **Count subarrays with sum divisible by K.**
Constraints: $1 \leq n \leq 10^5$

36. **Find smallest subarray covering all unique elements.**
Constraints: $1 \leq n \leq 10^5$
37. **Maximize sum of subarray after at most K swaps using two pointers.**
Constraints: $1 \leq n \leq 10^5$
38. **Sliding window median** — maintain median dynamically.
Constraints: $1 \leq n \leq 10^5$
39. **Longest substring containing at most K distinct vowels.**
Constraints: $1 \leq |s| \leq 10^5$
40. **Count subarrays where $\max - \min \leq K$.**
Constraints: $1 \leq n \leq 10^5$
41. **Longest subarray with sum divisible by K.**
Constraints: $1 \leq n \leq 10^5$
42. **Find smallest range covering elements from k sorted arrays.**
Constraints: $1 \leq n \leq 10^3, 1 \leq k \leq 10^3$
43. **Maximum length subarray with alternating even/odd elements.**
Constraints: $1 \leq n \leq 10^5$
44. **Longest arithmetic subarray using two pointers.**
Constraints: $1 \leq n \leq 10^5$
45. **Count number of subarrays with $\text{XOR} \leq K$.**
Constraints: $1 \leq n \leq 10^5$
46. **Find longest subarray with sum exactly K using sliding window.**
Constraints: $1 \leq n \leq 10^5$
47. **Maximize sum of non-overlapping subarrays of size k.**
Constraints: $1 \leq n \leq 10^5$
48. **Sliding window maximum / minimum problem.**
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
49. **Longest subarray with at most K changes to make all equal.**
Constraints: $1 \leq n \leq 10^5$
50. **Count number of subarrays with at most K odd numbers.**
Constraints: $1 \leq n \leq 10^5$

I can next create 10 Two-Pointer problems with input/output examples in Codeforces-style format, ready to implement and practice.

Do you want me to do that?

You said:

sliding window

ChatGPT said:

Here's a list of 50 Sliding Window problem statements with Codeforces-style constraints, suitable for Java, C++, or Python practice. ♠

◆ 50 Sliding Window Problems

□ Basic Level (1–10)

1. **Maximum sum subarray of size k.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 2. **Minimum sum subarray of size k.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 3. **Average of all subarrays of size k.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 4. **Count subarrays with sum exactly k** (non-negative array).
Constraints: $1 \leq n \leq 10^5$
 5. **Longest subarray with sum $\leq S$.**
Constraints: $1 \leq n \leq 10^5$
 6. **Longest subarray with sum exactly k** (non-negative array).
Constraints: $1 \leq n \leq 10^5$
 7. **Count subarrays with sum $\leq k$** (non-negative array).
Constraints: $1 \leq n \leq 10^5$
 8. **Maximum of all subarrays of size k** (Sliding Window Maximum).
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 9. **Minimum of all subarrays of size k** (Sliding Window Minimum).
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
 10. **Number of subarrays with sum $\geq S$** (non-negative array).
Constraints: $1 \leq n \leq 10^5$
-

□ Intermediate Level (11–30)

11. **Longest substring without repeating characters.**
Constraints: $1 \leq |s| \leq 10^5$
12. **Longest substring with at most k distinct characters.**
Constraints: $1 \leq |s| \leq 10^5$, $1 \leq k \leq 26$
13. **Longest substring containing at most k vowels.**
Constraints: $1 \leq |s| \leq 10^5$
14. **Count number of substrings with at most k distinct characters.**
Constraints: $1 \leq |s| \leq 10^5$
15. **Longest subarray containing at most k zeros.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
16. **Longest subarray containing at most k ones.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
17. **Count subarrays with exactly k odd numbers.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
18. **Longest subarray with at most k changes to make all equal.**
Constraints: $1 \leq n \leq 10^5$
19. **Maximum sum of subarray with at most k elements removed.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
20. **Longest substring containing exactly k distinct characters.**
Constraints: $1 \leq |s| \leq 10^5$

21. **Longest substring with at most k repeating characters allowed.**
Constraints: $1 \leq |s| \leq 10^5$
 22. **Longest substring with all unique vowels.**
Constraints: $1 \leq |s| \leq 10^5$
 23. **Count number of substrings with sum divisible by k (numeric string).**
Constraints: $1 \leq |s| \leq 10^5$
 24. **Find maximum average subarray of size k.**
Constraints: $1 \leq n \leq 10^5$
 25. **Sliding window product maximum/minimum.**
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
 26. **Longest subarray with difference between max and min $\leq k$.**
Constraints: $1 \leq n \leq 10^5$
 27. **Count number of subarrays with max - min $\leq k$.**
Constraints: $1 \leq n \leq 10^5$
 28. **Longest subarray with alternating even/odd numbers.**
Constraints: $1 \leq n \leq 10^5$
 29. **Longest arithmetic subarray (subarray with constant difference).**
Constraints: $1 \leq n \leq 10^5$
 30. **Maximum sum rectangle in 2D matrix (Kadane + sliding window).**
Constraints: $1 \leq n, m \leq 500$
-

● Advanced Level (31–50)

31. **Longest substring with at most k repeating letters.**
Constraints: $1 \leq |s| \leq 10^5$
32. **Count subarrays with exactly k distinct elements.**
Constraints: $1 \leq n \leq 10^5$
33. **Count number of substrings with exactly k vowels.**
Constraints: $1 \leq |s| \leq 10^5$
34. **Longest subarray where sum of even-indexed and odd-indexed elements $\leq k$.**
Constraints: $1 \leq n \leq 10^5$
35. **Maximum sum of non-overlapping subarrays of size k.**
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
36. **Sliding window median** — maintain median dynamically.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
37. **Longest subarray with sum divisible by k.**
Constraints: $1 \leq n \leq 10^5$
38. **Find smallest window containing all unique elements.**
Constraints: $1 \leq n \leq 10^5$
39. **Maximize sum of subarray after at most k swaps using sliding window.**
Constraints: $1 \leq n \leq 10^5$
40. **Longest substring with equal number of 0s and 1s.**
Constraints: $1 \leq |s| \leq 10^5$
41. **Count subarrays where XOR $\leq k$.**
Constraints: $1 \leq n \leq 10^5$
42. **Maximum sum of subarray with at most k negative numbers.**
Constraints: $1 \leq n \leq 10^5$

43. **Longest substring with at most k consonants.**
Constraints: $1 \leq |s| \leq 10^5$
44. **Longest substring with equal frequency of characters.**
Constraints: $1 \leq |s| \leq 10^5$
45. **Longest subarray with at most k increases/decreases allowed.**
Constraints: $1 \leq n \leq 10^5$
46. **Sliding window sum with updates (dynamic).**
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
47. **Longest substring with exactly k repeating characters.**
Constraints: $1 \leq |s| \leq 10^5$
48. **Longest subarray where all elements $\leq k$ after replacements.**
Constraints: $1 \leq n \leq 10^5$
49. **Count number of subarrays with sum in range [L, R].**
Constraints: $1 \leq n \leq 10^5$
50. **Maximum sum subarray after removing at most one element (Kadane + sliding window).**
Constraints: $1 \leq n \leq 10^5$

I can also **create 10 Sliding Window problems with sample input/output examples in Codeforces-style format**, ready for implementation.

Do you want me to do that next?

You said:

Binary search

ChatGPT said:

Here's a **list of 50 Binary Search / Binary Search on Answer problems** with **Codeforces-style constraints**, ranging from basic to advanced, suitable for Java, C++, or Python practice.



◆ 50 Binary Search Problems

□ Basic Level (1–10)

1. **Find an element in a sorted array.**
Constraints: $1 \leq n \leq 10^5, -10^9 \leq \text{arr}[i] \leq 10^9$
2. **Find the first occurrence of x in a sorted array.**
Constraints: $1 \leq n \leq 10^5$
3. **Find the last occurrence of x in a sorted array.**
Constraints: $1 \leq n \leq 10^5$

4. **Count occurrences of x in a sorted array.**
Constraints: $1 \leq n \leq 10^5$
 5. **Find the square root of a number** using binary search.
Constraints: $1 \leq n \leq 10^{18}$
 6. **Find the smallest number $\geq x$ in a sorted array.**
Constraints: $1 \leq n \leq 10^5$
 7. **Find the largest number $\leq x$ in a sorted array.**
Constraints: $1 \leq n \leq 10^5$
 8. **Search in a rotated sorted array (no duplicates).**
Constraints: $1 \leq n \leq 10^5$
 9. **Find the peak element** in an array (element greater than neighbors).
Constraints: $1 \leq n \leq 10^5$
 10. **Find the position to insert an element** in a sorted array.
Constraints: $1 \leq n \leq 10^5$
-

□ Intermediate Level (11–30)

11. **Binary search on answer: minimize maximum subarray sum.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
12. **Binary search on answer: minimum capacity to ship packages within D days.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^4$
13. **Find square root with decimal precision.**
Constraints: $1 \leq n \leq 10^{18}$, precision = 10^{-6}
14. **Aggressive cows / Largest minimum distance.**
Constraints: $2 \leq n \leq 10^5$, $1 \leq x[i] \leq 10^9$
15. **Minimum maximum element after dividing array into k parts.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
16. **Search in a rotated sorted array (with duplicates).**
Constraints: $1 \leq n \leq 10^5$
17. **Find fixed point ($\text{arr}[i] = i$) in sorted array.**
Constraints: $1 \leq n \leq 10^5$
18. **Find square root in integer array without using sqrt function.**
Constraints: $1 \leq n \leq 10^{18}$
19. **Find the smallest missing number in sorted array of distinct integers.**
Constraints: $1 \leq n \leq 10^5$
20. **Find peak in bitonic array** (first increasing then decreasing).
Constraints: $1 \leq n \leq 10^5$
21. **Find rotation count in rotated sorted array.**
Constraints: $1 \leq n \leq 10^5$
22. **Maximum average subarray of size k.**
Constraints: $1 \leq n \leq 10^5$
23. **Koko Eating Bananas** (minimum eating speed).
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$
24. **Allocate minimum number of pages** (binary search on answer).
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
25. **Capacity to ship packages** (split array sum $\leq \text{mid}$).
Constraints: $1 \leq n \leq 10^5$

26. **Find the smallest divisor such that sum of division \leq threshold.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
27. **Find square root in large numbers (binary search integer part).**
Constraints: $1 \leq n \leq 10^{18}$
28. **Find element closest to x in sorted array.**
Constraints: $1 \leq n \leq 10^5$
29. **Find first bad version** (classic LeetCode problem).
Constraints: $1 \leq n \leq 10^9$
30. **Search in 2D matrix** (matrix sorted row-wise and column-wise).
Constraints: $1 \leq n, m \leq 10^3$
-

● Advanced Level (31–50)

31. **Minimum largest sum among m subarrays** (LeetCode “Split Array Largest Sum”).
Constraints: $1 \leq n \leq 10^5$
32. **Find smallest radius to cover all houses with heaters.**
Constraints: $1 \leq n \leq 10^5$
33. **Allocate tasks to workers minimizing maximum workload.**
Constraints: $1 \leq n \leq 10^5$
34. **Minimize max distance to gas station** (floating-point binary search).
Constraints: $1 \leq n \leq 10^4$, precision = 10^{-6}
35. **Find minimum speed to complete journey within T hours.**
Constraints: $1 \leq n \leq 10^5$
36. **Find minimum time to produce m items** (factory problem).
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
37. **Find smallest x such that f(x) is true** (generic binary search on answer).
Constraints: $1 \leq n \leq 10^9$
38. **Maximum minimum distance between elements after removing k elements.**
Constraints: $1 \leq n \leq 10^5$
39. **Find square root to given precision for very large numbers.**
Constraints: $1 \leq n \leq 10^{18}$, precision = 10^{-9}
40. **Minimize maximum difference among k pairs.**
Constraints: $1 \leq n \leq 10^5$
41. **Aggressive cows in 2D coordinates** (binary search on distance).
Constraints: $1 \leq n \leq 10^5$
42. **Find minimum time to paint n boards** (painters partition problem).
Constraints: $1 \leq n \leq 10^5$
43. **Minimize maximum waiting time for customers** (queue allocation).
Constraints: $1 \leq n \leq 10^5$
44. **Find minimum capacity to deliver packages within D days** (sum \leq capacity).
Constraints: $1 \leq n \leq 10^5$
45. **Maximize minimum distance in placing antennas.**
Constraints: $1 \leq n \leq 10^5$
46. **Minimum largest element after dividing array into k parts** (binary search answer).
Constraints: $1 \leq n \leq 10^5$
47. **Find element in infinite sorted array** (simulate array with get(i)).
Constraints: $n \rightarrow \infty$

48. **Search in mountain array** (bitonic array).
Constraints: $1 \leq n \leq 10^5$
49. **Minimize maximum absolute difference between elements** in k subarrays.
Constraints: $1 \leq n \leq 10^5$
50. **Binary search on floating-point answer** (max distance / speed / precision problem).
Constraints: $1 \leq n \leq 10^5$, precision = 10^{-6}
-

I can next **create 10 Binary Search problems with sample input/output** in **Codeforces-style format**, ready to implement and practice.

Do you want me to do that?

You said:

prefix sum

ChatGPT said:

Here's a **list of 50 Prefix Sum / Cumulative Sum problem statements** with **Codeforces-style constraints**, ranging from basic to advanced, suitable for Java, C++, or Python practice.



◆ 50 Prefix Sum Problems

□ Basic Level (1–10)

1. **Compute prefix sum array** of a given array.
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
2. **Compute sum of elements from L to R using prefix sum.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
3. **Count number of elements in subarray equal to k** using prefix sum of frequency array.
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^5$
4. **Compute running average of array elements.**
Constraints: $1 \leq n \leq 10^5$
5. **Count number of even/odd elements in range [L, R].**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
6. **Cumulative sum modulo M** of array elements.
Constraints: $1 \leq n \leq 10^5$, $1 \leq M \leq 10^9$
7. **Find sum of all subarrays of size k** using prefix sum.
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
8. **Find sum of squares in subarray [L, R]** using prefix sums.
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^4$

9. **Count number of positive numbers in subarray [L, R].**

Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$

10. **Compute cumulative sum and print at each step.**

Constraints: $1 \leq n \leq 10^5$

□ Intermediate Level (11–30)

11. **Maximum sum subarray of size k** using prefix sum.

Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$

12. **Subarray sum divisible by k** (count using prefix sum modulo).

Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$

13. **Number of subarrays with sum exactly S** using prefix sum + hashmap.

Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^4$

14. **2D prefix sum / cumulative sum in matrix.**

Constraints: $1 \leq n, m \leq 10^3$

15. **Sum of all rectangles in 2D matrix queries.**

Constraints: $1 \leq n, m \leq 10^3$, $1 \leq q \leq 10^5$

16. **Number of subarrays with sum $\leq S$** in non-negative array.

Constraints: $1 \leq n \leq 10^5$

17. **Compute XOR of all subarrays** using prefix XOR.

Constraints: $1 \leq n \leq 10^5$

18. **Count number of subarrays with even sum** using prefix sum modulo 2.

Constraints: $1 \leq n \leq 10^5$

19. **Count number of subarrays with odd sum** using prefix sum modulo 2.

Constraints: $1 \leq n \leq 10^5$

20. **Find longest subarray with sum = 0** using prefix sum + hashmap.

Constraints: $1 \leq n \leq 10^5$, $-10^9 \leq \text{arr}[i] \leq 10^9$

21. **Range update + range sum query** using prefix sum technique.

Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$

22. **Difference array + prefix sum** for multiple range additions.

Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$

23. **Find maximum subarray sum** using prefix sum.

Constraints: $1 \leq n \leq 10^5$, $-10^9 \leq \text{arr}[i] \leq 10^9$

24. **Compute prefix sum of absolute values.**

Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$

25. **Compute prefix sum for string of digits.**

Constraints: $1 \leq |s| \leq 10^5$

26. **Count number of subarrays with sum multiple of k.**

Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$

27. **Range sum queries for dynamic array** using prefix sum.

Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$

28. **Maximum subarray sum in circular array** using prefix sums.

Constraints: $1 \leq n \leq 10^5$

29. **Minimum length subarray with sum $\geq S$** using prefix sum + binary search.

Constraints: $1 \leq n \leq 10^5$

30. **Prefix sum of multiples of k in array.**

Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$

● Advanced Level (31–50)

31. **2D cumulative XOR in matrix for queries.**
Constraints: $1 \leq n, m \leq 500, 1 \leq q \leq 10^5$
32. **Maximum sum rectangle in 2D matrix** using prefix sum + Kadane.
Constraints: $1 \leq n, m \leq 500$
33. **Count number of submatrices with sum = 0** using 2D prefix sum.
Constraints: $1 \leq n, m \leq 500$
34. **Range addition + prefix sum query in large array** (difference array technique).
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
35. **Sum of all subarrays of length $\geq k$.**
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
36. **Maximum sum of subarray after at most one deletion.**
Constraints: $1 \leq n \leq 10^5$
37. **Number of subarrays with sum in range [L, R].**
Constraints: $1 \leq n \leq 10^5, |\text{arr}[i]| \leq 10^9$
38. **Prefix sum of weighted array** (weight = index * value).
Constraints: $1 \leq n \leq 10^5$
39. **Count number of subarrays with sum divisible by k in 2D matrix.**
Constraints: $1 \leq n, m \leq 500, 1 \leq k \leq 10^9$
40. **Maximum average subarray of size $\geq k$** using prefix sums + binary search.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq n$
41. **Number of subarrays with sum $\leq S$** using prefix sum + sliding window.
Constraints: $1 \leq n \leq 10^5$
42. **Dynamic prefix sum with updates** (Fenwick tree / BIT implementation).
Constraints: $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$
43. **Sum of diagonal rectangles in 2D matrix** using prefix sum.
Constraints: $1 \leq n, m \leq 500$
44. **Maximum sum submatrix with at most K elements removed.**
Constraints: $1 \leq n, m \leq 500$
45. **Count number of subarrays with positive sum** using prefix sum.
Constraints: $1 \leq n \leq 10^5$
46. **Prefix sum of differences array** ($\text{arr}[i+1] - \text{arr}[i]$).
Constraints: $1 \leq n \leq 10^5$
47. **Maximum sum subarray modulo M** using prefix sums + binary search.
Constraints: $1 \leq n \leq 10^5, 1 \leq M \leq 10^9$
48. **Sum of all odd-length subarrays** using prefix sum.
Constraints: $1 \leq n \leq 10^5$
49. **2D prefix sum for range updates in matrix.**
Constraints: $1 \leq n, m \leq 500, 1 \leq q \leq 10^5$
50. **Count number of subarrays where sum = multiple of k** in large array.
Constraints: $1 \leq n \leq 10^5, 1 \leq k \leq 10^9$

I can next create 10 Prefix Sum problems with input/output examples in Codeforces-style format, ready to implement and practice.

Do you want me to do that?

You said:

back tracking

ChatGPT said:

Here's a list of **50 Backtracking problem statements** with **Codeforces/LeetCode-style constraints**, ranging from easy to advanced, suitable for Java, C++, or Python practice. ♠

◆ 50 Backtracking Problems

□ Basic Level (1–10)

1. **Generate all subsets of a given set.**
Constraints: $1 \leq n \leq 16$
 2. **Generate all permutations of a string.**
Constraints: $1 \leq |s| \leq 10$
 3. **Generate all permutations of an array of n distinct numbers.**
Constraints: $1 \leq n \leq 10$
 4. **Print all subsets whose sum = k (Subset Sum).**
Constraints: $1 \leq n \leq 20$, $1 \leq \text{arr}[i] \leq 100$
 5. **Generate all binary strings of length n .**
Constraints: $1 \leq n \leq 16$
 6. **Generate all palindromic partitions of a string.**
Constraints: $1 \leq |s| \leq 16$
 7. **Print all combinations of r elements from n elements.**
Constraints: $1 \leq n \leq 16$, $1 \leq r \leq n$
 8. **Generate all subsets with distinct elements.**
Constraints: $1 \leq n \leq 16$
 9. **Print all strings formed from a given set of characters of length n .**
Constraints: $1 \leq n \leq 10$, $|\text{charset}| \leq 10$
 10. **Generate all permutations with duplicates allowed.**
Constraints: $1 \leq n \leq 10$
-

□ Intermediate Level (11–30)

11. **Solve N-Queens problem** (place n queens on $n \times n$ board).
Constraints: $1 \leq n \leq 12$
12. **Rat in a Maze problem** (find all paths from top-left to bottom-right).
Constraints: $1 \leq n$, $m \leq 10$

13. **Word Search** (existence of word in grid).
Constraints: $1 \leq n, m \leq 10, |\text{word}| \leq 10$
 14. **Sudoku Solver** (fill a 9×9 board).
Constraints: 9×9 board
 15. **Generate all valid parentheses sequences of length $2n$.**
Constraints: $1 \leq n \leq 12$
 16. **Combination Sum** (all unique combinations that sum to target).
Constraints: $1 \leq n \leq 20, 1 \leq \text{target} \leq 100$
 17. **Combination Sum II** (each number used once).
Constraints: $1 \leq n \leq 20, 1 \leq \text{target} \leq 100$
 18. **Letter Combinations of a Phone Number** (map digits to letters).
Constraints: $1 \leq |\text{digits}| \leq 10$
 19. **Palindrome Partitioning II** (all possible palindrome partitions).
Constraints: $1 \leq |s| \leq 16$
 20. **Restore IP Addresses** from string of digits.
Constraints: $4 \leq |s| \leq 12$
 21. **Permutations II** (handle duplicates in array).
Constraints: $1 \leq n \leq 10$
 22. **Find all subsets with sum divisible by k .**
Constraints: $1 \leq n \leq 20, 1 \leq \text{arr}[i] \leq 100$
 23. **Generate all words of length n with given vowels/consonants rules.**
Constraints: $1 \leq n \leq 10$
 24. **Find all ways to climb stairs with 1, 2, or 3 steps.**
Constraints: $1 \leq n \leq 20$
 25. **Find all sequences of dice throws summing to target.**
Constraints: $1 \leq n \leq 10, 1 \leq \text{target} \leq 30$
 26. **Knight's Tour problem** (visit all squares of chessboard).
Constraints: $n = 5 \sim 8$
 27. **Subset sum with negative numbers allowed.**
Constraints: $1 \leq n \leq 20, |\text{arr}[i]| \leq 50$
 28. **Print all k -length sequences from n numbers without repetition.**
Constraints: $1 \leq n \leq 10, 1 \leq k \leq n$
 29. **Find all partitions of integer n into sum of positive integers.**
Constraints: $1 \leq n \leq 20$
 30. **Generate all strings avoiding consecutive repeated characters.**
Constraints: $1 \leq n \leq 10, |\text{charset}| \leq 10$
-

● Advanced Level (31–50)

31. **N-Knights problem** (place knights on $n \times n$ board without attacking).
Constraints: $1 \leq n \leq 10$
32. **M Coloring problem** (graph coloring with backtracking).
Constraints: $1 \leq n \leq 10, 1 \leq m \leq n$
33. **Hamiltonian Path problem** in small graph.
Constraints: $1 \leq n \leq 10$
34. **Find all magic squares of order n .**
Constraints: $3 \leq n \leq 4$

35. **Find all solutions for cryptarithmic puzzles** (SEND + MORE = MONEY).
Constraints: Letters ≤ 10
36. **All possible expressions evaluating to target** (add operators +, -, *, /).
Constraints: $1 \leq |\text{digits}| \leq 10$, target ≤ 1000
37. **Count number of unique paths in grid with obstacles** using backtracking.
Constraints: $1 \leq n, m \leq 10$
38. **Rat in a 3D Maze problem** (3D grid paths).
Constraints: $1 \leq n, m, k \leq 5$
39. **All valid Sudoku boards** (generate all solutions).
Constraints: 9×9 board
40. **All possible words from phone keypad with constraints.**
Constraints: $1 \leq |\text{digits}| \leq 10$
41. **All combinations of coins to sum target** (infinite supply).
Constraints: $1 \leq n \leq 20$, $1 \leq \text{target} \leq 100$
42. **All combinations of coins to sum target** (limited supply).
Constraints: $1 \leq n \leq 20$, $1 \leq \text{target} \leq 100$
43. **All arrangements of numbers to satisfy inequality constraints.**
Constraints: $1 \leq n \leq 10$
44. **Word Ladder all shortest transformation sequences.**
Constraints: $|\text{word}| \leq 5$, $n \leq 100$
45. **Print all possible knight moves sequences of length k.**
Constraints: $1 \leq n \leq 8$, $1 \leq k \leq 10$
46. **All subsets with sum in range [L, R].**
Constraints: $1 \leq n \leq 20$
47. **Find all balanced bracket sequences with multiple types of brackets.**
Constraints: $1 \leq n \leq 10$
48. **All solutions to the 8-Queens problem.**
Constraints: $n = 8$
49. **Find all ways to partition array into k subsets.**
Constraints: $1 \leq n \leq 10$, $1 \leq k \leq n$
50. **All arrangements of numbers forming a magic square.**
Constraints: $n \leq 3$

I can also **create 10 Backtracking problems with input/output examples in Codeforces-style format**, ready to implement and practice.

Do you want me to do that next?

You said:

dynamic programing

ChatGPT said:

Here's a **list of 50 Dynamic Programming (DP) problem statements** with **Codeforces/LeetCode-style constraints**, ranging from basic to advanced, suitable for Java, C++, or Python practice. ♣

◆ 50 Dynamic Programming Problems

□ Basic Level (1–10)

1. **Fibonacci number (nth term).**
Constraints: $1 \leq n \leq 10^5$
2. **Climbing stairs** (1 or 2 steps).
Constraints: $1 \leq n \leq 10^5$
3. **Min cost climbing stairs** (array of costs).
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{cost}[i] \leq 10^4$
4. **Longest Increasing Subsequence (LIS)** using DP.
Constraints: $1 \leq n \leq 10^3$, $|\text{arr}[i]| \leq 10^4$
5. **Longest Common Subsequence (LCS)** of two strings.
Constraints: $1 \leq |s1|, |s2| \leq 10^3$
6. **0/1 Knapsack problem** (maximize value with weight limit).
Constraints: $1 \leq n \leq 100$, $1 \leq W \leq 10^3$
7. **Minimum number of coins to make amount** (coin change problem).
Constraints: $1 \leq n \leq 50$, $1 \leq \text{amount} \leq 10^4$
8. **Count number of ways to make amount** (coin change combinations).
Constraints: $1 \leq n \leq 50$, $1 \leq \text{amount} \leq 10^4$
9. **Maximum sum subarray** (Kadane's Algorithm / DP).
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
10. **Edit distance between two strings.**
Constraints: $1 \leq |s1|, |s2| \leq 500$

□ Intermediate Level (11–30)

11. **Maximum sum increasing subsequence.**
Constraints: $1 \leq n \leq 10^3$, $|\text{arr}[i]| \leq 10^4$
12. **Number of ways to reach nth stair with steps 1,2,3.**
Constraints: $1 \leq n \leq 10^5$
13. **Subset sum problem** (check if sum exists).
Constraints: $1 \leq n \leq 100$, $1 \leq \text{sum} \leq 10^3$
14. **Partition equal subset sum.**
Constraints: $1 \leq n \leq 100$, $1 \leq \text{arr}[i] \leq 100$
15. **Longest Palindromic Subsequence.**
Constraints: $1 \leq n \leq 500$
16. **Longest Palindromic Substring using DP.**
Constraints: $1 \leq n \leq 1000$
17. **Paint House problem** (minimum cost to paint n houses with k colors).
Constraints: $1 \leq n \leq 1000$, $1 \leq k \leq 20$
18. **Maximum product subarray.**
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^4$

19. **Count number of binary strings without consecutive 1s.**
Constraints: $1 \leq n \leq 10^5$
 20. **Minimum jumps to reach end of array.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^3$
 21. **Matrix chain multiplication (min cost).**
Constraints: $1 \leq n \leq 100$
 22. **Egg dropping problem (min trials).**
Constraints: $1 \leq n \leq 100$, $1 \leq k \leq 100$
 23. **Rod cutting problem (maximize profit).**
Constraints: $1 \leq n \leq 100$
 24. **Count ways to tile $2 \times n$ board using 1×2 and 2×1 tiles.**
Constraints: $1 \leq n \leq 10^5$
 25. **Distinct subsequences (number of times s_2 appears in s_1).**
Constraints: $1 \leq |s_1|, |s_2| \leq 1000$
 26. **Min cost path in matrix (right, down, diagonally).**
Constraints: $1 \leq n, m \leq 500$
 27. **Max sum path in matrix (only right/down).**
Constraints: $1 \leq n, m \leq 500$
 28. **Maximum length chain of pairs.**
Constraints: $1 \leq n \leq 10^3$, $1 \leq \text{pair}[i] \leq 10^4$
 29. **Palindrome partitioning (min cuts).**
Constraints: $1 \leq n \leq 500$
 30. **Weighted job scheduling (maximize profit).**
Constraints: $1 \leq n \leq 10^4$, $1 \leq \text{profit}[i] \leq 10^4$
-

● Advanced Level (31–50)

31. **Russian Doll Envelopes (max envelopes nested).**
Constraints: $1 \leq n \leq 10^3$
32. **Max sum rectangle in 2D matrix using DP.**
Constraints: $1 \leq n, m \leq 500$
33. **Maximum sum submatrix no larger than K.**
Constraints: $1 \leq n, m \leq 100$
34. **Longest Bitonic Subsequence.**
Constraints: $1 \leq n \leq 10^3$
35. **Min cost to cut a stick into pieces (DP).**
Constraints: $1 \leq n \leq 100$
36. **Burst Balloons (maximize coins).**
Constraints: $1 \leq n \leq 500$
37. **Minimum falling path sum in matrix.**
Constraints: $1 \leq n, m \leq 500$
38. **Max sum path with obstacles in grid using DP.**
Constraints: $1 \leq n, m \leq 500$
39. **Distinct subsequences with modulo (large numbers).**
Constraints: $1 \leq n \leq 10^3$
40. **Count number of ways to decode a numeric string.**
Constraints: $1 \leq |s| \leq 10^5$

41. **Max coins from picking array elements with neighbors removed** (House Robber problem).
Constraints: $1 \leq n \leq 10^5$
42. **Max coins in circular array** (circular House Robber).
Constraints: $1 \leq n \leq 10^5$
43. **Longest repeating subsequence.**
Constraints: $1 \leq |s| \leq 500$
44. **Wildcard pattern matching** (DP with ?, *).
Constraints: $1 \leq |s|, |p| \leq 500$
45. **Count distinct ways to climb stairs modulo 10^9+7 .**
Constraints: $1 \leq n \leq 10^5$
46. **Minimum number of insertions to make string palindrome.**
Constraints: $1 \leq |s| \leq 500$
47. **Max path sum in binary tree** (tree DP).
Constraints: $1 \leq n \leq 10^5$
48. **Min cost to reach last cell in weighted grid** using DP.
Constraints: $1 \leq n, m \leq 500$
49. **Max sum increasing path in matrix.**
Constraints: $1 \leq n, m \leq 500$
50. **Count number of ways to partition integer n into sums.**
Constraints: $1 \leq n \leq 1000$

I can next create 10 Dynamic Programming problems with sample input/output examples in Codeforces-style format, ready for implementation.

Do you want me to do that next?

You said:

greedy Algorithm

ChatGPT said:

Here's a list of 50 Greedy Algorithm problem statements with Codeforces/LeetCode-style constraints, ranging from basic to advanced, suitable for Java, C++, or Python practice.



◆ 50 Greedy Algorithm Problems

□ Basic Level (1–10)

1. **Activity Selection Problem** – select maximum non-overlapping activities.
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{start}[i] < \text{end}[i] \leq 10^9$

2. **Fractional Knapsack Problem** – maximize value with fractional items.
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{weight}[i]$, $\text{value}[i] \leq 10^9$
 3. **Minimum number of coins to make amount** (if coins unlimited).
Constraints: $1 \leq n \leq 100$, $1 \leq \text{amount} \leq 10^4$
 4. **Maximum sum of absolute differences in array.**
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
 5. **Job sequencing with deadlines** (maximize profit).
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{deadline}[i]$, $\text{profit}[i] \leq 10^9$
 6. **Huffman Coding / Minimum cost of ropes.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
 7. **Maximum number of events one can attend** (sorted by end time).
Constraints: $1 \leq n \leq 10^5$
 8. **Minimize sum of differences by pairing numbers.**
Constraints: $1 \leq n \leq 10^5$, n even, $|\text{arr}[i]| \leq 10^9$
 9. **Largest number formed from array of numbers.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq \text{arr}[i] \leq 10^9$
 10. **Minimize the maximum height difference when dividing items.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$
-

□ Intermediate Level (11–30)

11. **Assign tasks to workers minimizing total completion time.**
Constraints: $1 \leq n \leq 10^5$
12. **Fractional job allocation / maximize profit.**
Constraints: $1 \leq n \leq 10^5$
13. **Gas station / minimum refills to reach destination.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{distance}[i] \leq 10^9$
14. **Minimum number of platforms required for trains.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arrival}[i]$, $\text{departure}[i] \leq 10^9$
15. **Interval covering problem** – minimum number of points to cover all intervals.
Constraints: $1 \leq n \leq 10^5$, $1 \leq l[i] \leq r[i] \leq 10^9$
16. **Minimum number of arrows to burst balloons.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq x[i]$, $y[i] \leq 10^9$
17. **Minimize waiting time / completion time of tasks.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{duration}[i] \leq 10^9$
18. **Largest sum of subsequence with no adjacent elements.**
Constraints: $1 \leq n \leq 10^5$
19. **Minimum coins to reach target with limited denominations.**
Constraints: $1 \leq n \leq 100$, $1 \leq \text{amount} \leq 10^4$
20. **Maximum meetings in one room** (sorted by start/end).
Constraints: $1 \leq n \leq 10^5$
21. **Activity selection with start time priority.**
Constraints: $1 \leq n \leq 10^5$
22. **Maximize sum of numbers after deleting k elements.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
23. **Minimum cost to connect ropes.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{arr}[i] \leq 10^9$

- 24. **Maximize stock profit with unlimited transactions.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq \text{price}[i] \leq 10^9$
 - 25. **Minimize product of two arrays after rearranging.**
Constraints: $1 \leq n \leq 10^5$, $|\text{arr}[i]| \leq 10^9$
 - 26. **Fractional assignments of tasks to maximize efficiency.**
Constraints: $1 \leq n \leq 10^5$
 - 27. **Assign cookies to children** (maximize satisfied children).
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{size}[i]$, $\text{greed}[i] \leq 10^9$
 - 28. **Maximum points by picking cards from ends.**
Constraints: $1 \leq n \leq 10^5$, $|\text{points}[i]| \leq 10^4$
 - 29. **Minimize cost of connecting sticks.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{sticks}[i] \leq 10^9$
 - 30. **Select intervals with maximum coverage.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq l[i] \leq r[i] \leq 10^9$
-

● Advanced Level (31–50)

- 31. **Minimum number of conference rooms for overlapping intervals.**
Constraints: $1 \leq n \leq 10^5$
- 32. **Minimize maximum difference in array after k operations.**
Constraints: $1 \leq n \leq 10^5$
- 33. **Maximize number of envelopes nested** (Russian Doll Envelopes).
Constraints: $1 \leq n \leq 10^5$
- 34. **Maximize number of tasks completed before deadline.**
Constraints: $1 \leq n \leq 10^5$
- 35. **Fractional knapsack with multiple constraints.**
Constraints: $1 \leq n \leq 10^5$
- 36. **Maximize score by merging stones.**
Constraints: $1 \leq n \leq 10^3$
- 37. **Minimum cost to hire workers for tasks.**
Constraints: $1 \leq n \leq 10^5$
- 38. **Minimize total waiting time for customers** (like ATM problem).
Constraints: $1 \leq n \leq 10^5$
- 39. **Maximum profit from stock transactions with cooldown.**
Constraints: $1 \leq n \leq 10^5$
- 40. **Minimize sum of absolute differences** after rearranging two arrays.
Constraints: $1 \leq n \leq 10^5$
- 41. **Maximum sum of elements in k disjoint intervals.**
Constraints: $1 \leq n \leq 10^5$
- 42. **Minimum number of coins with special constraints.**
Constraints: $1 \leq n \leq 10^5$
- 43. **Optimal task scheduling with penalties.**
Constraints: $1 \leq n \leq 10^5$
- 44. **Select maximum number of intervals with limited resources.**
Constraints: $1 \leq n \leq 10^5$
- 45. **Maximize number of points in game with greedy picks.**
Constraints: $1 \leq n \leq 10^5$

46. **Minimize max distance to supply** (like antenna placement).
Constraints: $1 \leq n \leq 10^5$
 47. **Maximize profit by selling products with deadlines**.
Constraints: $1 \leq n \leq 10^5$
 48. **Minimize number of lectures to cover topics**.
Constraints: $1 \leq n \leq 10^5$
 49. **Minimize total time in scheduling jobs on machines**.
Constraints: $1 \leq n \leq 10^5$
 50. **Maximize number of tasks done under weight/capacity constraint**.
Constraints: $1 \leq n \leq 10^5$
-

I can next create 10 Greedy Algorithm problems with sample input/output examples in Codeforces-style format, ready to implement and practice.

Do you want me to do that next?

You said:

intervals

ChatGPT said:

Here's a list of 50 Interval-related problem statements (covering merging, scheduling, covering, and optimization) with Codeforces/LeetCode-style constraints, ranging from basic to advanced, suitable for Java, C++, or Python practice. ♠

◆ 50 Interval Problems

□ Basic Level (1–10)

1. **Merge overlapping intervals**.
Constraints: $1 \leq n \leq 10^5$, $0 \leq \text{start}[i] \leq \text{end}[i] \leq 10^9$
2. **Insert a new interval into a list of sorted intervals**.
Constraints: $1 \leq n \leq 10^5$
3. **Check if a new interval overlaps with any existing interval**.
Constraints: $1 \leq n \leq 10^5$
4. **Interval intersection between two lists of intervals**.
Constraints: $1 \leq n, m \leq 10^5$
5. **Remove covered intervals** (intervals completely inside others).
Constraints: $1 \leq n \leq 10^5$
6. **Maximum number of non-overlapping intervals** (activity selection).
Constraints: $1 \leq n \leq 10^5$

7. **Minimum number of intervals to remove to make rest non-overlapping.**
Constraints: $1 \leq n \leq 10^5$
 8. **Check if intervals can be arranged without overlap.**
Constraints: $1 \leq n \leq 10^5$
 9. **Count number of overlapping intervals at any point.**
Constraints: $1 \leq n \leq 10^5$
 10. **Find gaps between intervals (interval complement).**
Constraints: $1 \leq n \leq 10^5$
-

□ Intermediate Level (11–30)

11. **Maximum length covered by merged intervals.**
Constraints: $1 \leq n \leq 10^5$
12. **Minimum number of points to cover all intervals.**
Constraints: $1 \leq n \leq 10^5$
13. **Check if intervals form a continuous coverage from L to R.**
Constraints: $1 \leq n \leq 10^5$
14. **Find the interval with maximum overlap count.**
Constraints: $1 \leq n \leq 10^5$
15. **Insert multiple intervals and merge all.**
Constraints: $1 \leq n, m \leq 10^5$
16. **Sort intervals by start or end time.**
Constraints: $1 \leq n \leq 10^5$
17. **Find the longest chain of intervals where each interval starts after previous ends.**
Constraints: $1 \leq n \leq 10^5$
18. **Minimum intervals to remove for complete coverage.**
Constraints: $1 \leq n \leq 10^5$
19. **Check if intervals cover a certain point or range.**
Constraints: $1 \leq n \leq 10^5$
20. **Maximum number of intervals covering a given point.**
Constraints: $1 \leq n \leq 10^5$
21. **Count disjoint sets of intervals.**
Constraints: $1 \leq n \leq 10^5$
22. **Find median interval length.**
Constraints: $1 \leq n \leq 10^5$
23. **Merge k sorted lists of intervals.**
Constraints: $1 \leq k \leq 10^3$, total intervals $\leq 10^5$
24. **Check if any intervals overlap in a list.**
Constraints: $1 \leq n \leq 10^5$
25. **Maximum number of intervals to select under weight constraint.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{weight}[i] \leq 10^9$
26. **Count intervals overlapping with query interval.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
27. **Find earliest interval ending after a given start.**
Constraints: $1 \leq n \leq 10^5$
28. **Minimize sum of lengths of chosen intervals to cover all points.**
Constraints: $1 \leq n \leq 10^5$

29. **Maximum sum of weights of non-overlapping intervals.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq \text{weight}[i] \leq 10^9$
30. **Maximum number of intervals starting before a certain time.**
Constraints: $1 \leq n \leq 10^5$
-

● Advanced Level (31–50)

31. **Weighted interval scheduling (maximize total weight).**
Constraints: $1 \leq n \leq 10^5$
32. **Find interval with maximum coverage after removing k intervals.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
33. **Find union length of multiple intervals.**
Constraints: $1 \leq n \leq 10^5$
34. **Count number of intervals fully contained within each interval.**
Constraints: $1 \leq n \leq 10^5$
35. **Check if a point lies in majority of intervals.**
Constraints: $1 \leq n \leq 10^5$
36. **Find the k -th point in merged intervals.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq 10^9$
37. **Dynamic insertion and deletion of intervals, count overlaps.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
38. **Interval covering problem (minimum intervals covering all points).**
Constraints: $1 \leq n \leq 10^5$
39. **Maximum gap between intervals after merging.**
Constraints: $1 \leq n \leq 10^5$
40. **Intervals intersection with query interval.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
41. **Find the smallest interval covering a given set of points.**
Constraints: $1 \leq n \leq 10^5$
42. **Merge overlapping intervals online (real-time).**
Constraints: $1 \leq n \leq 10^5$
43. **Find interval that overlaps with maximum number of other intervals.**
Constraints: $1 \leq n \leq 10^5$
44. **Select maximum weighted intervals without overlap (weighted activity selection).**
Constraints: $1 \leq n \leq 10^5$
45. **Minimize cost of interval coverage with penalties for uncovered gaps.**
Constraints: $1 \leq n \leq 10^5$
46. **Find first interval fully containing a given query interval.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$
47. **Split intervals into minimum number of non-overlapping groups.**
Constraints: $1 \leq n \leq 10^5$
48. **Find maximum number of intervals that intersect at a common point.**
Constraints: $1 \leq n \leq 10^5$
49. **Count number of points covered by at least k intervals.**
Constraints: $1 \leq n \leq 10^5$, $1 \leq k \leq n$
50. **Find interval with maximum overlap after shifting all intervals by d units.**
Constraints: $1 \leq n \leq 10^5$, $0 \leq d \leq 10^9$

