

Question 4:(Easy Sum of sub sets)

Problem Statement: A gaming company is analyzing character performance in its latest strategy game. Each character (hero) has a strength value represented by an integer. To simulate various battle scenarios, the game engine needs to compute the overall combat effectiveness of all possible teams that can be formed. The power of a team is calculated using the formula:

$(\text{maximum strength in the team})^2 \times (\text{minimum strength in the team})$.

Given the array of hero strengths, your task is to develop a system that computes the total power of all possible non-empty combinations of heroes (teams). Since the number of combinations and resulting total power can be extremely large, the final result should be returned modulo $10^9 + 7$. This metric will be used by the game developers to balance team dynamics, improve matchmaking algorithms, and adjust the difficulty levels based on team composition diversity.

Constraints:

$1 \leq \text{nums.length} \leq 10^5$

$1 \leq \text{nums}[i] \leq 10^9$

Example 1:

Input: nums = [2,1,4]

Output: 141

Explanation:

1st group: [2] has power = $2^2 * 2 = 8$.

2nd group: [1] has power = $1^2 * 1 = 1$.

3rd group: [4] has power = $4^2 * 4 = 64$.

4th group: [2,1] has power = $2^2 * 1 = 4$.

5th group: [2,4] has power = $4^2 * 2 = 32$.

6th group: [1,4] has power = $4^2 * 1 = 16$.

7th group: [2,1,4] has power = $4^2 * 1 = 16$.

The sum of powers of all groups is $8 + 1 + 64 + 4 + 32 + 16 + 16 = 141$.

Example 2:

Input: nums = [1,1,1]

Output: 7

Explanation: A total of 7 groups are possible, and the power of each group will be 1.

Therefore, the sum of the powers of all groups is 7.

Input0:

1 4 8

Output0:

977

Input1:

3 4 2

Output1:

229

Input2:

3 4 2 1

Output2:

316

Input3:

1 2

Output3:

13

Question 5:(parity optimization-Completely uncovered topic)

Problem Statement: A cybersecurity company is analyzing patterns in network traffic logs represented as character streams. Each log entry is a long string of characters where each character represents a type of request or signal. To detect anomalies or specific behavioral patterns, analysts want to study the imbalance between frequently and infrequently occurring signals within certain time windows. Specifically, for any continuous segment (substring) of the log with a minimum length of k , they want to find the maximum difference in frequency between two signal types, where one occurs an odd number of times and the other an even number of times.

- subs has a size of at least k .
- Character a has an odd frequency in subs.
- Character b has an even frequency in subs.
- Return the maximum difference.

This metric helps identify irregular communication bursts or suspicious traffic patterns that might indicate system misuse or threats. Given the character stream and the integer k , your task is to compute the highest such frequency difference. This insight will aid in enhancing the company's anomaly detection systems.

Constraints:

$3 \leq s.length \leq 3 * 10^4$

s consists only of digits '0' to '4'.

The input is generated that at least one substring has a character with an even frequency and a character with an odd frequency.

$1 \leq k \leq s.length$

Example 1:

Input: $s = "12233"$, $k = 4$

Output: -1

Explanation:

For the substring "12233", the frequency of '1' is 1 and the frequency of '3' is 2. The difference is $1 - 2 = -1$.

Example 2:

Input: $s = "1122211"$, $k = 3$

Output: 1

Explanation:

For the substring "11222", the frequency of '2' is 3 and the frequency of '1' is 2. The difference is $3 - 2 = 1$.

Example 3:

Input: $s = "110"$, $k = 3$

Output: -1

Input0:

1223

4

Output0:

-1

Input1:
1122211
4
Output1:
1

Input2:
33222
4
Output2:
1

Input3:
1112
2
Output3:
-1

Question 6:

Problem Statement: You are working as a software engineer for a financial analytics company that builds tools to detect unusual market behavior. One of your tasks is to analyze a series of stock price changes over several days. These changes are represented as an array of integers, where each integer indicates the change in stock price for a given day—positive for a gain and negative for a loss.

To identify patterns of consistent risk, your team defines the "beauty" of a sliding window of days as the x -th smallest negative change in that period. If there are fewer than x negative changes in a window, the beauty is considered 0.

You are asked to write a function that, given an array of daily price changes, a window size k , and a rank x , returns a list of beauties for all subarrays of size k .

Your solution will help identify high-risk trends in historical price data.

Constraints:
 $n == \text{nums.length}$
 $1 \leq n \leq 10^5$
 $1 \leq k \leq n$
 $1 \leq x \leq k$
 $-50 \leq \text{nums}[i] \leq 50$

Example 1:

Input: $\text{nums} = [1, -1, -3, -2, 3]$, $k = 3$, $x = 2$

Output: $[-1, -2, -2]$

Explanation: There are 3 subarrays with size $k = 3$.

The first subarray is $[1, -1, -3]$ and the 2nd smallest negative integer is -1 .

The second subarray is $[-1, -3, -2]$ and the 2nd smallest negative integer is -2 .

The third subarray is $[-3, -2, 3]$ and the 2nd smallest negative integer is -2 .

Example 2:

Input: nums = [-1,-2,-3,-4,-5], k = 2, x = 2

Output: [-1,-2,-3,-4]

Explanation: There are 4 subarrays with size k = 2.

For [-1, -2], the 2nd smallest negative integer is -1.

For [-2, -3], the 2nd smallest negative integer is -2.

For [-3, -4], the 2nd smallest negative integer is -3.

For [-4, -5], the 2nd smallest negative integer is -4.

Example 3:

Input: nums = [-3,1,2,-3,0,-3], k = 2, x = 1

Output: [-3,0,-3,-3,-3]

Explanation: There are 5 subarrays with size k = 2.

For [-3, 1], the 1st smallest negative integer is -3.

For [1, 2], there is no negative integer so the beauty is 0.

For [2, -3], the 1st smallest negative integer is -3.

For [-3, 0], the 1st smallest negative integer is -3.

For [0, -3], the 1st smallest negative integer is -3.

Input0:

1 -1 -3 -2 3 4

3

2

Output0:

-1 -2 -2 0

Input1:

1 -1 -3 -2 3 -4

3

2

Output1:

-1 -2 -2 -2

Input2:

-1 -3 -2 3 -4

3

3

Output2:

-1 0 0

Input3:

-1 -3 -2 3 -4

2

2

Output3:

-1 -2 0 0