

All-Ukrainian Collegiate Programming Contest  
2023  
I stage  
Розбір задач

Дякуємо за участь!

## Задача А. Напишіть чекер

Якщо  $answer = a + b$ , виведіть `Accepted`. В іншому випадку виведіть `Wrong Answer`.

## Задача В. Фіксовані точки

Для кожної позиції  $pos$  спробуємо порахувати кількість підмасивів  $p$ , в яких елемент  $p_{pos}$  не зміниться після сортування.

Розглянемо підмасив  $p[l : r]$  для  $l \leq pos \leq r$ .  $p_{pos}$  залишиться на місці після його сортування тоді й лише тоді, коли в ньому рівно  $pos - l$  чисел, строго менших за  $p_{pos}$ .

Для кожного  $i \leq pos$  порахуємо  $pref_i$ : кількість чисел в відрізку  $p[i : pos]$ , що строго менші за  $p_{pos}$ . Також, для кожного  $i \geq pos$  порахуємо  $suf_i$ : кількість чисел в відрізку  $p[pos : i]$ , що строго менші за  $p_{pos}$ . Тепер, для кожного  $l \leq pos$  достатньо знайти кількість позицій  $r \geq pos$ , для яких  $suf_r = (pos - l) - pref_l$ . Ми можемо знайти, скільки разів кожне число зустрічається в масиві  $suf$ , за час  $O(n)$ , а тому й загальний час на обробку однієї позиції  $pos$  —  $O(n)$ .

Загальна асимптотика:  $O(n^2)$ .

## Задача С. Залишкові конфігурації

Очевидно, якщо  $a_i = a_j$  на початку,  $a_i$  завжди буде рівним  $a_j$ , тому викинемо з масиву дуплікати. Також, без обмеження загальності, можна вважати, що  $a_1 > a_2 > \dots > a_n$ .

Припустимо, що після виконання кількох операцій ми отримали масив  $[b_1, b_2, \dots, b_n]$ . Якщо  $a_1 = b_1$ , то ми жодного разу не вибирали  $x \leq a_1$ , а отже жодне число не змінилось. Припустимо тепер, що  $b_1 < a_1$ . Нехай  $a_1 - b_1 = y$ .

Покажемо, що для всіх  $i$  таких, що  $a_i \geq y$ , виконується  $a_i - b_i = y$ . Покажемо, що для такого  $i$   $a_1 - a_i$  не змінюється впродовж операцій. Доведемо це від супротивного: розглянемо перший момент, коли  $a_1 - a_i$  змінюється, тобто  $a_1 \bmod x - a_i \bmod x \neq a_1 - a_i$ .

Помітимо, що  $x > b_1 = a_1 - y \geq a_1 - a_i$  (перша нерівність виконується, бо інакше  $a_1$  обов'язково стало б меншим за  $b_1$ ). Якщо  $a_1 \bmod x \geq a_i \bmod x$ , то  $a_1 \bmod x - a_i \bmod x = a_1 - a_i$ , адже  $a_1 \bmod x - a_i \bmod x$  і  $a_1 - a_i$  мають однакову остачу при діленні на  $x$  та обидва лежать в проміжку  $[0, x - 1]$ . Тому ми маємо  $a_1 \bmod x < a_i \bmod x$ . Але якби  $a_1 \bmod x$  було не меншим за  $a_1 - a_i$ ,  $a_i \bmod x$  було б рівним  $a_1 \bmod x - (a_1 - a_i) < a_1 \bmod x$ , тому  $a_1 \bmod x < a_1 - a_i$ . Але  $b_1 \geq a_1 - a_i$ ! Суперечність.

Таким чином, ми маємо мати  $b_i = a_i - y$  для всіх  $i$ , для яких  $a_i \geq y$ . Це еквівалентно до застосування операції з  $x = y$  з самого початку. Єдине обмеження: ми маємо мати  $y > \frac{a_1}{2}$ , адже інакше  $a_1 \bmod y < a_1 - y$ . А далі ми просто розв'язуємо задачу для всіх чисел, менших за  $y$ , з єдиною умовою, що ми не можемо вибирати  $x \leq b_1$ .

Тепер задачу нескладно розв'язати за допомогою динамічного програмування:  $dp[pos][lim]$  — кількість різних конфігурацій, яку можна отримати з масиву  $a[pos : n]$ , якщо ми можемо обирати лише  $x \geq lim$ . Ми можемо робити переходи за  $O(\max(MAX, n))$ , просто перебираючи фінальне значення  $b_{pos}$ .

Загальна асимптотика:  $O(nMAX \cdot \max(n, MAX))$ , де  $MAX$  позначає максимальний елемент в масиві.

*Коментар:* автор задачі підозрює, що її можна оптимізувати до  $O(\max(n, MAX)^2)$ , але залишить це як вправу.

## Задача D. XOR > AND

Якщо всі числа рівні, скажімо, числу  $x$ , то XOR будь-якої підмножини не перевищує  $x$ , а AND будь-якої підмножини рівний  $x$ , тому таке розбиття не існуватиме.

Якщо ж не всі з них рівні, то покладемо максимальне число в першу групу, а всі інші числа — в іншу. XOR чисел в першій групі буде рівний максимальному числу в  $a$ , а AND чисел в другій групі не перевищуватиме мінімального числа в  $a$ . Отже, нерівність справджуватиметься.

Асимптотика:  $O(n)$ .

## Задача Е. Неспадний

Розв'яжемо задачу для конкретного  $x$ . Якщо масив вже неспадний, ми не маємо робити жодних операцій. Інакше, ми маємо замінити якісь елементи масиву на  $x$ , щоб він став неспадним.

Розглянемо, як виглядатиме масив після застосування даних операцій: в ньому буде неспадний префікс (можливо, пустий) з чисел  $< x$ , потім деякий відрізок з чисел  $x$ , потім неспадний суфікс з чисел  $> x$ . Наше завдання — замінити на  $x$  якомога меншу кількість чисел, тому ми хочемо максимізувати суму довжин цих префіксу та суфіксу. Щоб максимізувати префікс, потрібно знайти максимальне число  $i$ , для якого  $a_1 \leq a_2 \leq \dots \leq a_i < x$ . Щоб максимізувати суфікс, потрібно знайти мінімальне число  $i$ , для якого  $x < a_i \leq a_{i+1} \leq \dots \leq a_n$ .

Помітимо, що ці максимальні префікс та суфікс не можуть перетинатись, а тому достатньо знайти їх максимально можливі довжини незалежно одне від одного. Ми можемо знайти обидва значення за  $O(n)$ , просто йдучи зліва направо для префіксу та справа наліво для суфіксу. Кількість операцій буде рівна  $n - (\text{довжина префікса}) - (\text{довжина суфікса}) - (\text{кількість разів, яку } x \text{ зустрічається в масиві})$ .

Тепер розв'яжемо задачу для кожного  $x$  від 1 до  $n$ . Спершу порахуємо кількість разів, яку кожне число зустрічається в масиві  $a$ . Лишилось для кожного  $x$  знайти максимальне значення  $i$ , що  $a_1 \leq a_2 \leq \dots \leq a_i < x$  (а потім зробити те ж саме для суфіксів).

Помітимо, що ці значення можуть лише зростати при збільшенні  $x$ . Тому ми все ще можемо знайти їх за  $O(n)$ : перебираємо  $x$  від 1 до  $n$ , для кожного йдемо вправо, поки відповідний префікс все ще є неспадним та всі його елементи менші за  $x$ .

Загальна асимптотика:  $O(n)$ .

## Задача F. Антипаліндромний

Якщо якийсь символ  $x$  зустрічається в рядку рівно один раз, то відповідь YES: ми можемо поставити його першим, а інші символи розставити в довільному порядку. Надалі вважаємо, що кожен символ, що присутній в рядку, зустрічатиметься в ньому принаймні двічі.

Якщо в рядку всі символи рівні, то  $r_1 r_2$  обов'язково буде паліндромом, тому відповідь NO.

Якщо в рядку рівно два різні символи, то відповідь також NO. Для цього поглянемо на друге входження в рядок  $r$  символу  $r_1$ , скажімо, в позиції  $pos$ . Тоді рядок  $r_1 r_2 \dots r_{pos}$  є паліндромом.

Якщо ж в рядку принаймні три різні символи, то виберемо один з них,  $x$ . Поставимо перше входження  $x$  на початку, а всі інші в кінці. Інші ж символи поставимо між ними у відсортованому порядку: вони гарантовано не формуватимуть паліндром. Легко бачити, що така конструкція задовольняє умовам задачі.

Асимптотика  $O(n)$ .

## Задача G. Конфігурації мінімумів

Якщо для кожного  $i$   $a_i = b_i$ , то умова задачі очевидно виконується. Інакше, розглянемо мінімальне  $x$ , для якого існує таке  $pos$ , що  $a_{pos} \neq b_{pos}$ , і  $\min(a_{pos}, b_{pos}) = x$ . Без обмеження загальності,  $a_{pos} = x < b_{pos}$ .

Розглянемо довільну позицію  $i \neq pos$ , для неї має виконуватись  $\min(a_{pos}, b_i) = \min(b_{pos}, a_i)$ . Є два варіанти, як це може статись:

- Якщо  $\min(a_{pos}, b_i) = \min(b_{pos}, a_i) = x$ , то маємо  $a_i = x, b_i \geq x$ .
- Якщо  $\min(a_{pos}, b_i) = \min(b_{pos}, a_i) < x$ , то маємо  $a_i = b_i < x$ .

Таким чином, при фіксованому  $x$  масив  $a$  однозначно відновлюватиметься за масивом  $b$ : якщо  $b_i < x$ , то  $a_i = b_i$ , інакше  $a_i = x$ . При цьому, має існувати принаймні одна позиція  $pos$ , в якій  $b_i > x$ . Легко переконатись, що такі масиви задовольнятимуть умовам задачі.

Порахуємо кількість таких масивів для фіксованого  $x$ . Вона рівна  $m^n - x^n$ : загальній кількості можливих масивів  $b$  мінус кількість масивів  $b$ , всі елементи яких не перевищують  $x$ .

Таким чином, загальна відповідь:  $m^n + 2 \cdot \sum_{x=1}^{n-1} (m^n - x^n)$ .

Загальна асимптотика:  $O(m \log n)$ , якщо рахувати кожен степінь за допомогою швидкого піднесення до степеня.

## Задача Н. Різні суми

Покажемо, що в оптимальній конструкції немає підмасивів довжини  $\geq 5$ . Дійсно, припустимо, що в нас є  $i$ -ий підмасив  $[a_l, a_{l+1}, \dots, a_r]$  з  $r - l + 1 \geq 5$ . Розглянемо всі  $r - l$  способів розрізати цей підмасив на два підмасиви. Оскільки конструкція оптимальна, кожен з цих способів не задовольняє умові задачі.

З іншого боку, кожна з умов  $s_{i-1} \neq s_i, s_i \neq s_{i+1}, s_{i+1} \neq s_{i+2}$  не буде виконуватись максимум при одному розрізі, суперечність. Отже, в оптимальній конструкції всі підмасиви мають довжину  $\leq 4$ .

Тепер ми можемо розв'язати задачу за допомогою динамічного програмування. Нехай  $dp[pos][len]$  — максимальна кількість підмасивів, на яку можна розрізати  $a[1 : pos]$ , щоб суми кожних двох сусідніх підмасивів були різні, а довжина останнього підмасиву була рівно  $len$ . Ми можемо порахувати чергове  $dp[pos][len]$  за  $O(1)$ , перебираючи довжину передостаннього підмасиву.

Всього станів  $O(n)$ , тому загальна асимптотика також  $O(n)$ .

*Коментар:* автору невідомо, чи правда, що завжди є оптимальна конструкція, в якій довжини всіх відрізків не більші за 3, але це рішення проходить всі тести.

## Задача I. Максимальні суми підмасивів

Для кожного масиву  $arr_i$  порахуємо три значення:  $pref_i, suf_i, ans_i$ : найбільшу суму на префіксі, найбільшу суму на суфіксі, і найбільшу суму на підвідрізку відповідно. Ми можемо порахувати всі ці значення за  $O(|arr_i|)$ .

Легко бачити, що  $f(concat(arr_i, arr_j))$  рівний  $\max(ans_i, ans_j, suf_i + pref_j)$ . Дійсно, підвідрізок з максимальною сумою або лежатиме повністю всередині  $arr_i$ , або повністю всередині  $arr_j$ , або складатиметься з якогось суфіксу  $arr_i$  та якогось префіксу  $arr_j$ . Ми могли б обчислити суму цих значень за  $O(n^2)$ , але це занадто повільно.

Впорядкуємо масиви  $arr_i$  за зростанням  $ans_i$ , тобто відтепер вважаємо, що  $ans_1 \leq ans_2 \leq \dots \leq ans_n$ . Тоді для  $i < j$  ми маємо  $f(concat(arr_i, arr_j)) = \max(ans_i, ans_j, suf_i + pref_j) = \max(ans_j, suf_i + pref_j)$ .

Порахуємо суму всіх цих значень по  $i < j$ . Для фіксованого  $j$ , в суму  $\sum_{i=1}^j \max(ans_j, suf_i + pref_j)$  входить  $ans_j$  при  $suf_i \leq ans_j - pref_j$ , і  $suf_i + pref_j$  інакше. Таким чином, щоб порахувати  $\sum_{i=1}^j \max(ans_j, suf_i + pref_j)$ , нам достатньо дізнатись кількість  $i < j$ , для яких  $suf_i > ans_j - pref_j$ , і суму  $suf_i$  по всіх таких  $i$ .

Розв'язуватимемо більш загальну задачу: потрібно побудувати структуру даних, яка підтримує наступні операції:

- Додати число  $x$  до мультимножини  $S$ .
- Вивести кількість чисел в  $S$ , більших за  $y$ .
- Вивести суму чисел в  $S$ , більших за  $y$ .

Скористаймось тим, що ми знаємо значення всіх  $x$ , що ми додаватимемо до  $S$ . "Стиснемо" їх, занумерувавши їх від 1 до  $n$  в порядку зростання; позначатимемо відповідне значення від 1 до  $n$  як  $tar[x]$ . Тепер ми можемо просто використовувати два дерева відрізків. В одному ми зберігатимемо кількості доданих чисел, в іншому суму: при черговому запиті додавання  $x$ , ми додаватимемо 1 в позиції  $tar[x]$  в одному з них, а в іншому в цій же позиції додаватимемо  $x$ .

Аналогічно знайдемо суму  $f(concat(arr_i, arr_j))$  по  $i \geq j$ .

Загальна асимптотика:  $O(n \log n)$ .

## Задача J. Максимізуйте префіксні максимуми

Спершу спробуємо знайти якусь оцінку зверху на дану суму. Якщо  $f((q_1, q_2, \dots, q_n)) + f((p_{q_1}, p_{q_2}, \dots, p_{q_n})) = n + k$  для якогось  $k$ , то має бути принаймні  $k$  позицій  $pos$  таких, що  $q_{pos} = \max(q_1, q_2, \dots, q_{pos})$  і  $p_{q_{pos}} = \max(p_{q_1}, p_{q_2}, \dots, p_{q_{pos}})$ . Помітимо, що ці позиції утворюють зростаючу підпослідовність в  $p$ : якщо ці позиції  $pos_1, pos_2, \dots, pos_k$ , то ми маємо:

- $q_{pos_1} < q_{pos_2} < \dots < q_{pos_k}$
- $p_{q_{pos_1}} < p_{q_{pos_2}} < \dots < p_{q_{pos_k}}$

Таким чином, відповідь не може перевищувати  $n + LIS$ , де  $LIS$  — довжина найдовшої зростаючої підпослідовності перестановки  $p$ . Покажемо, що  $n + LIS$  досягається.

Для полегшення подальшого розуміння розв'язання подивимось на задачу трішки з іншого боку: в нас є пари  $(i, p_i)$ , і ми хочемо розставити ці пари в якомусь порядку так, щоб максимізувати суму кількості префіксних максимумів в перестановці з перших чисел в парі, і в перестановці з других чисел в парі. Для пари  $a$  позначатимемо перше число в парі як  $a^1$ , а друге як  $a^2$ .

Розглянемо підпослідовність зростаючих пар довжини  $LIS$ , розставимо їх в порядку зростання, нехай це пари  $a_1, a_2, \dots, a_{pos}$ . Наше завдання — розставити інші  $n - LIS$  пар так, щоб в кожній з них перше чи друге число було префіксним максимумом, а в вибраних  $LIS$  парах кожне число з пари було префіксним максимумом.

Для кожної іншої пари  $b$  знайдемо найбільше  $i$ , для якого  $a_i^1 < b^1$ , чи  $a_i^2 < b^2$ . Будемо ставити пару  $b$  між  $a_i$  та  $a_{i+1}$ , а саме: спершу поставимо всі пари  $b$ , для яких  $b^1 > a_i^1$ , в порядку зростання  $b^1$ , а потім всі пари  $b$ , для яких  $b^2 > a_i^2$ . В кожній з цих пар один з елементів буде префіксним максимумом.

Ми можемо знайти  $LIS$  за  $O(n \log n)$ . Загальна асимптотика:  $O(n \log n)$ .

## Задача К. Максимізуйте LCS перестановок

Розіб'ємо перестановку на цикли: послідовності чисел  $b_1, b_2, \dots, b_k$  такі, що  $p_{b_1} = b_2, p_{b_2} = b_3, \dots, p_{b_k} = b_1$ . Легко бачити, що, якщо довжина циклу більша за 1, то не знайдеться такої перестановки  $q$ , для якої елементи циклу в  $q$  йшли б в тому ж порядку, що й в перестановці  $r = (p_{q_1}, p_{q_2}, \dots, p_{q_n})$ . Дійсно, елементи циклу знаходяться в  $p$  та  $r$  на одних й тих самих позиціях, тому, якщо вони знаходяться в них в однаковому порядку, ми маємо мати  $q_i = p_{q_i}$  для кожного  $q_i$  з циклу, але це неможливо.

Отже, з кожного такого циклу довжини  $k \geq 2$  ми можемо взяти не більше ніж  $k - 1$  елементів в кожену спільну підпослідовність. Покажемо, що це досягається.

Для кожного циклу по черзі, додамо до  $q$   $b_1, b_2, \dots, b_k$  в цьому порядку. До  $r$  ми додамо  $p_{b_1}, p_{b_2}, \dots, p_{b_k} = b_2, b_3, \dots, b_1$ . Після цього, ми можемо додати числа  $b_2, b_3, \dots, b_k$  до спільної підпослідовності  $q$  та  $r$ ! (А у випадку  $k = 1$ , ми додаємо до неї один елемент  $b_1$ ).

Отже, розв'язок наступний: виписати всі цикли по порядку. Загальна асимптотика:  $O(n)$ .

## Задача L. Не зростаючий

Якщо масив вже не є зростаючим, то нам не потрібно робити жодних операцій, і відповідь 0.

Інакше, ми маємо  $a_1 < a_2 < \dots < a_n$ , і ми хочемо отримати  $a_i \geq a_{i+1}$  для якогось  $1 \leq i \leq n-1$ . Для фіксованого  $i$  мінімальна кількість операцій, необхідна для цього, рівна  $a_{i+1} - a_i$ , адже необхідно зробити  $a_i$  не меншим за  $a_{i+1}$ .

Таким чином, відповідь — це мінімальне значення по  $a_{i+1} - a_i$  по  $1 \leq i \leq n-1$ .

Асимптотика:  $O(n)$ .