Задача A. MMM

Нехай $b_i = a_{i+1} - a_i$. Тоді нам потрібно зробити так, щоб b_i спочатку були додатними, а потім від'ємними.

Тоді під час операції у нас збільшиться одне b_i та зменшиться інше b_i. Тепер нам потрібно розв'язати іншу задачу:

1. Можна вибрати (l, r) $(1 \leqslant l < r < n)$. Виконати $b_l = b_l + 1$; $b_r = b_r - 1$.
2. Можна вибрати l $(1 \leqslant l < n)$. Виконати $b_l = b_l + 1$.
3. Можна вибрати r $(1 \leqslant r < n)$. Виконати $b_r = b_r - 1$.

Нехай у нас є певне зафіксоване k. Тоді відповідь

$$c_k = \max(\sum_{j=1}^{k-1} \max(1 - b_j, 0), \sum_{j=k}^{n-1} \max(1 + b_j, 0))$$

Таке можна порахувати за $O(n^2)$.

Можемо помітити, що формула складається з двох частин. Давайте випишемо формулу для першої частини.

$$L_k = \sum_{j=1}^{k-1} \max(1 - b_j, 0)$$

$$L_k = L_{k-1} + \max(1 - b_{k-1}, 0)$$

Ось таке ми можемо порахувати за $O(n)$. Аналогічно з R_k.

Тоді виходить, що $c_k = \max(L_k, R_k)$.

Задача B. Сніг

Давайте знайдемо два числа: L та R — максимальні здвиги вліво та вправо відповідно. Можна помітити, що вони будуть однаковими для всіх шарів.

Будемо розглядати кожну пару сусідніх шарів. Будемо розглядати лише відрізок між ними. Тоді певний префікс відійде до лівого шару, а деяких суфікс відійде до правого. Нехай відстань d. Якщо $d \geqslant L + R$, тоді лівому відійде рівно R, а правому рівно L. Якщо ж $d < L + R$, тоді нам потрібно знайти першим момент часу, коли $d < L + R$, якщо операція була здвигу вправо, то відповіді будуть $(d - L, L)$, а якщо вліво, то $(R, d - R)$.

Для кожної пари можна знайти відповідь за $O(q)$, всього пар $O(n)$, тому складність $O(nq)$.

Можемо спочатку перебирати дні, а потім в i-ий день знаходити відповіді для тих пар шарів, у яких $d < L+R$. Тобто ми можемо відсортувати пари шарів за зростанням відстані, а потім поступово знаходити відповіді. Таке рішення має складність $O(q + n \log q)$.

Задача C. Свап

Наша відповідь буде складатися з підмасивів, де кожний підмасив - це послідовність кількох перших чисел, що залишилися, у спадаючому порядку. Наприклад, $[3, 2, 1, 5, 4]$. Тут два підмасиви — $[3, 2, 1]$ та $[5, 4]$.

Буде знаходити $dp[i]$ — відповідь, якщо ми використали перші i елементів на перших i позиціях. Тоді в кожного $dp[i]$ буде $O(n)$ переходів, і кожен перехід ми можемо порахувати за $O(n^2)$. Сумарна складність $O(n^4)$.

Розглянемо перехід з $dp[j]$ в $dp[k]$. Розглянемо усі числа h $(j < h \leqslant k)$. Знайдемо i таке, що $a_i = h$. Тоді кількість операцій, які потрібно для того, щоб перемістити число h на правильну позицію, буде рівно сумі

- Кількістю таких t, що $t < i$ та $a_t > k$.

- Кількістю таких t, що $t < i$ та $j < a_t \leqslant h$.

Можемо препроцесингом порахувати $c[i][j]$ — кількість чисел серед перших i, які більші за j. Це можна порахувати за $O(n^2)$. Маючи такий масив, можна для певного h порахувати відповідь як $c[i][j] - c[i][h] + c[i][k]$. Сумарна складність тепер $O(n^3)$.

Коли ми порахували перехід з $dp[j]$ в $dp[k]$, ми можемо запам'ятати ціну такого переходу. Після чого перейти до переходу з $dp[j+1]$ в $dp[k]$. Ціна сильно не зміниться. Різницю цін можна порахувати за $O(1)$.

Задача D. Місто і робот

Помітимо, що якщо ми змінюємо колір, то нам вигідно завжди змінювати на той колір, який ще не використовується у графі.

Помітимо, що якщо нам потрібно перейти з однієї вершини в сусідню, то є два варіанти: або перефарбувати це ребро, або перефарбувати усі ребра цього кольору.

Побудуємо новий граф (вершина; колір ребро, по якому перейшли останній раз). Якщо ребро було перефарбоване, то 0. Додамо ребра та запустимо алгоритм Дейкстри.