Задача A. Разрез прямоугольника

Пусть $a \leqslant b$. Рассмотрим несколько случаев:

- Если a чётно, то мы можем разрезать прямоугольник на два прямоугольника размера $\frac{a}{2} \times b$ и сложить из них прямоугольник $\frac{a}{2} \times 2b$, который точно отличается от $a \times b$.
- Если b чётно и $b \neq 2a$, то мы можем разрезать прямоугольник на два прямоугольника размера $a \times \frac{b}{2}$ и сложить из них прямоугольник размера $2a \times \frac{b}{2}$. Заметим, что здесь мы пользуемся тем, что $b \neq 2a$, так как если $b = 2a$, то мы получим такой же прямоугольник размера $b \times a$.
- Если a и b нечётные или $b = 2a$ и a нечётно, то прямоугольник не является интересным. Легко понять, что если мы разрежем прямоугольник размера $a \times b$ на два прямоугольника размера $a \times c$ и $a \times d$, где $c \neq d$, то мы всегда сможем сложить только исходный прямоугольник (аналогично если мы разрежем на прямоугольники $c \times b$ и $d \times b$). А отсюда следует, что мы должны разделить одну из сторон прямоугольника пополам, поэтому хотя бы одна сторона должна быть чётной.

Пусть $x = \lceil \frac{n}{2} \rceil$ — количество нечётных чисел от 1 до n. Тогда количество интересных прямоугольников можно вычислить следующим образом: всего есть $\frac{n(n+1)}{2}$ различных прямоугольников, длины сторон которых не превосходят n. Вычтем из этого количества количество прямоугольников, у которых обе стороны нечётны. Это количество равно $\frac{x(x+1)}{2}$. После этого нужно вычесть те прямоугольники, у которых большая сторона равна удвоенной меньшей, и при этом меньшая сторона нечётная. Это количество равно $\left\lceil \frac{\lfloor \frac{n}{2} \rfloor}{2} \right\rceil$, потому что в таком случае меньшая сторона не должна превышать $\lfloor \frac{n}{2} \rfloor$, и при этом должна быть нечётной.

Получаем итоговую формулу: $\frac{n(n+1)}{2} - \frac{x(x+1)}{2} - \left\lceil \frac{\lfloor \frac{n}{2} \rfloor}{2} \right\rceil$.

Задача B. Урок физкультуры

Все номера повторяются через $2k - 2$ позиции. Если у мальчика Васи номер при расчете равен x, то он может быть на позициях либо вида $(2k-2) \cdot t + x$, либо $(2k-2) \cdot t + k + k - x$, для каких-то неотрицательных t. Это верно, для всех x, кроме $x = 1$ и $x = k$ — для этих значений остается только один вариант.

Давайте зафиксируем один из вариантов, для второго все будет аналогично. Нам нужно найти сколько различных k удовлетворяют равенству $(2k-2) \cdot t + x = n$, при некотором неотрицательном t. Несложно видеть, что это выполняется тогда и только тогда, когда $n - x$ делится на $2k - 2$. Поэтому нужно найти число **чётных** делителей числа $n - x$. Чтобы рассмотреть второй случай нужно поступить аналогично с числом $n + x - 2$. Асимптотика решения: $O(\sqrt{n})$

Задача C. Подземелья Одинокой горы

Научимся решать задачу при $n = 1$. Пусть есть только одна раса и количество её представителей равно c. Заметим, что при фиксированном k нам выгодно разделить представителей расы по отрядам практически поровну.

Если c кратно k, то нам выгодно в каждый отряд взять ровно по $y = \frac{c}{k}$ существ. И тогда суммарное количество пар существ, которые находятся в разных отрядах, равно $\frac{k(k-1)}{2} \cdot y^2$ (всего есть $\frac{k(k-1)}{2}$ пар отрядов, и для каждой пары отрядов есть y^2 пар существ из разных отрядов).

В общем случае, когда c может быть не кратно k, обозначим $y = \lfloor \frac{c}{k} \rfloor$ и $y' = \lceil \frac{c}{k} \rceil$. Тогда нам выгодно сделать отряды размера y и y', причём количество отрядов размера y' равно $c \bmod k$ (мы как бы делаем все отряды размера y, а потом в какие-то отряды добавляем по 1 из оставшейся части). В таком случае суммарное количество пар существ, которые находятся в разных отрядах, равно $C^2_{k - c \bmod k} \cdot y^2 + C^2_{c \bmod k} \cdot y'^2 + (k - c \bmod k) \cdot (c \bmod k) \cdot y \cdot y'$. Осталось заметить, что нет смысла делать $k > c$, поэтому можно просто перебрать k от 1 до c и выбрать оптимальное.

При $n > 1$ можно заметить, что при фиксированном k мы можем решать задачу независимо для каждой расы. Пусть количество представителей i-й расы равно c_i. Тогда переберём для неё k от 1 до c_i и прибавим максимальную суммарную силу к значению cnt_k (массив cnt является общим для всех рас). Также заметим, что при $k > c_i$ мы получим такую же суммарную силу, как и при $k = c_i$. Тогда в дополнительном массиве add (опять же общем для всех рас) прибавим к add_{c_i} максимальную суммарную силу для $k = c_i$.

Получаем такое решение: сначала посчитаем описанные массивы cnt и add. После этого переберём k от 1 до максимального c_i. Максимальная суммарная сила отрядов для фиксированного k будет равна $(cnt_k + (\text{сумма значений } add_i \text{ для } i < k)) \cdot b - (k-1) \cdot X$. Из данных значений нужно выбрать максимум.

Задача D. Модообразная последовательность

Посмотрим, как будет выглядеть ответ: сначала будет идти префикс вида $x, x+y, \ldots, x+k\cdot y$, а после — какое-то число блоков вида $x \bmod y, x \bmod y + y, \ldots, x \bmod y + k \cdot y$.

Мы можем вычесть из всех элементов последовательности число $x \bmod y$, а после разделить все элементы на y (все элементы будут делиться на y, так как изначально у них был остаток $x \bmod y$). Пусть $b_1 = \frac{x - x \bmod y}{y}$. Тогда наша последовательность будет начинаться с $b_1, b_1+1, \ldots, b_1+k_1$, а после будут идти блоки вида $0, 1, \ldots, k_i$.

Посчитаем такие значения: dp_i — минимальная длина последовательности из блоков вида $0, 1, \ldots, k_j$, имеющая сумму i. Можно посчитать это значение для всех чисел от 0 до S методом динамического программирования. Если мы обработали все значения от 0 до $k-1$, то для k мы посчитали минимальную длину, и мы можем обновить значение dp для $k+1, k+1+2, \ldots$ — всего $O(\sqrt{S})$ значений, не превышающих S. В этом же dp можно сохранить, через какие значения мы пересчитывались, для восстановления ответа.

Теперь, мы можем перебрать длину первого блока вида $b_1, b_1+1, \ldots, b_1+k_1$. Тогда мы знаем сумму оставшихся блоков, и с помощью предпосчитанного dp узнаем, можно ли составить искомую последовательность или нет.

Задача E. Цифровые узоры

Предположим, что $a_i = a_{i+1}$ для какого-то $1 \leqslant i < n$, тогда при любом $1 \leqslant j \leqslant m$ клетки (i, j) и $(i+1, j)$ будут иметь одинаковую прозрачность. Аналогичное утверждение можно сделать если нашелся индекс j: $b_j = b_{j+1}$.

Тогда позиции $a_i = a_{i+1}$ делят массив a на *блоки*, в каждом из которых все пары соседей не равны друг другу. При чем понятно, что если нашелся квадрат (x, y, d) состоящий из клеток (i, j) таких, что $x \leqslant i < x+d$ и $y \leqslant j < y+d$, то отрезок $[x, x+d-1]$ находится целиком в одном из этих *блоков* массива a. Аналогично на блоки можно поделить массив b, и тогда отрезок $[y, y+d-1]$ тоже будет целиком лежать в одном из блоков.

Попробуем решить задачу за $O(1)$, если в массиве a и b нет соседних элементов с одинаковыми значениями (также предположим, что $n \leqslant m$):

$$f(n,m) = \sum_{k=1}^{n}(n-k+1)(m-k+1) = \sum_{k=1}^{n}\left(k^2 + (m-n)k\right) f(n,m) = \frac{n(n+1)(2n+1)}{6} + (m-n)\cdot\frac{n(n+1)}{2}$$

Эту формулу можно дополнительно преобразовать, введя для каждого натурального n четверку чисел $a_n = 1$, $b_n = n$, $c_n = \frac{1}{2}n(n+1)$, $d_n = \frac{1}{6}n(n+1)(2n+1) - \frac{1}{2}n^2(n+1)$. Тогда $f(n,m) = d_n a_m + c_n b_m$, если $n \leqslant m$ и $f(n,m) = a_n d_m + b_n c_m$, если $n > m$.

Но если все же в массивах a и b есть соседние одинаковые элементы, то это значит, что они как-то разделены на блоки. Если это блоки длины $n_1, \ldots, n_k$ в массиве a и блоки длины $m_1, \ldots, m_l$ в массиве b, то ответ на задачу — это

$$\text{ans} = \sum_{i=1}^{k}\sum_{j=1}^{l} f(n_i, m_j)$$

Научимся быстро считать суммы вида $f(x, m_1) + \ldots + f(x, m_l)$. Для этого заведем 4 дерева отрезков для того, чтобы быстро вычислять суммы $\sum a_y$, $\sum b_y$, $\sum c_y$, $\sum d_y$ на отрезках по y, с учетом кратности y в массиве $m_1, \ldots, m_l$. Теперь вычисление $f(x, m_1) + \ldots + f(x, m_k)$ свелов к 4 запросам ДО:

$$f(x, m_1) + \ldots + f(x, m_l) = a_x \cdot \sum_{m_i < x} d_{m_i} + b_x \cdot \sum_{m_i < x} c_{m_i} + c_x \cdot \sum_{m_i \geqslant x} b_{m_i} + d_x \cdot \sum_{m_i \geqslant x} a_{m_i}$$

Сумма $f(n_1, y) + \ldots + f(n_k, y)$ считается аналогично. Теперь осталось собрать наше решение в кучу. Будем в режиме онлайн поддерживать блоки массивов a и b. Это очень удобно делать, если хранить позиции $a_i = a_{i+1}$ в структуре по типу `std::set`, а также если работать с разностным массивом a (то есть поддерживать не сам массив a, а массив разностей соседних элементов $c_i = a_{i+1} - a_i$). Чтобы пересчитывать ответ будем считать количество квадратов, которые участвуют в конкретном блоке массива a **или** b, пользуясь результатом выше. В итоге получили решение за $O((n+q)(\log n + \log m))$.

P.S. Решение за $O(q\sqrt{n})$ не зайдет в виду большой константы. Я очень постарался его отсечь :D.

Задача F. Суммы модулей

Рассмотрим какое-нибудь $x \in [0, c]$, и пусть $f(a, x) = A$ и $f(b, x) = B$.

Предположим, что $A \leqslant B$. Тогда для всех $i \leqslant A$ выполняется $a_i, b_i \leqslant x$, для всех $i \in [A+1, B]$ выполняется $b_i \leqslant x < a_i$ и для всех $i \in [B+1, n]$ выполняется $x < a_i, b_i$. Отсюда можно увидеть, что величина $B - A$ равна количеству таких $i \in [1, n]$, что $x \in [b_i, a_i - 1]$.

Проведя аналогичное рассуждение для $A > B$, получаем, что $|f(a, x) - f(b, x)|$ равно количеству таких $i \in [1, n]$, что $x \in [\min(a_i, b_i), \max(a_i, b_i) - 1]$. Фиксированное $i \in [1, n]$ будет посчитано по всем x суммарно $|a_i - b_i|$ раз, поэтому функцию $g(a, b, c)$ можно записать как $g(a, b, c) = \sum_{i=1}^{n} |a_i - b_i|$.

Таким образом, нам нужно посчитать сумму величин $\sum_{i=1}^{n} |a'_i - b'_i|$ по всем парам последовательностей a', b', подходящих под шаблон. Будем считать сумму независимо для каждого $i \in [1, n]$.

Так как в a и b нет двух подряд идущих -1, сумма $|a'_i - b'_i|$ не зависит от выбора a'_j, b'_j для $j \neq i$, поэтому можно посчитать s_i — сумму $|a'_i - b'_i|$ в предположении, что a'_j и b'_j зафиксированы для всех $j \neq i$ и w_i — количество способов корректно выбрать a'_i и b'_i, в предположении, что a'_j и b'_j зафиксированы для всех $j \neq i$. Также нужно посчитать W — общее количество пар a', b', подходящих под шаблон. (отдельно нужно разобрать случай $W = 0$) После этого сумма $|a'_i - b'_i|$ по всем последовательностям будет равна $s_i \cdot \frac{W}{w_i} = W \cdot \frac{s_i}{w_i}$. (для обработки запросов изменения позднее будет полезно сначала посчитать сумму $\frac{s_i}{w_i}$ по всем i) и только потом умножить ее на W.

Для того, чтобы посчитать s_i, нам нужно научиться считать сумму $|x - y|$ по всем $x \in [l_x, r_x]$ и $y \in [l_y, r_y]$ для каких-то отрезков $[l_x, r_x]$ и $[l_y, r_y]$. Обозначим эту величину как $s(l_x, r_x, l_y, r_y)$.

Значение $|x - y|$ равно количеству целых $t \in [\min(x, y), \max(x, y) - 1]$. Тогда вместо того, чтобы считать сумму $|x - y|$ по всем x, y, посчитаем сумму по всем t количества пар x, y, таких, что $t \in [\min(x, y), \max(x, y) - 1]$. Для удобства будем считать отдельно пары, где $x \leqslant t < y$ и пары, где $y \leqslant t < x$. Обозначим через $s_1(l_x, r_x, l_y, r_y)$ сумму по всем $t \in [0, c]$ количеств пар $x \in [l_x, r_x]$, $y \in [l_y, r_y]$, таких, что $x \leqslant t < y$. Тогда $s(l_x, r_x, l_y, r_y) = s_1(l_x, r_x, l_{y,r} y) + s_1(l_y, r_y, l_x, r_x)$.

Научимся считать $s_1(l_x, r_x, l_y, r_y)$. Количество $x \in [l_x, r_x]$, таких, что $x \leqslant t$ равно $\max(0, \min(r_x, t) - l_x + 1)$. Количество $y \in [l_y, r_y]$, таких, что $y > t$, равно $\max(0, r_y - \max(l_y, t+1) + 1)$. Если мы будем рассматривать только $t \in [l_x, r_y - 1]$, то отрезки подходящих x и y всегда будут непустыми, поэтому можно убрать взятие максимума с нулем. Итого, мы хотим посчитать сумму

$$\sum_{t=l_x}^{r_y - 1} (\min(r_x, t) - l_x + 1)(r_y - \max(l_y, t+1) + 1)$$

При $t \leqslant r_x$ выполняется $\min(r_x, t) = t$, а при $t > r_x$ выполняется $\min(r_x, t) = r_x$. Аналогично, при $t \leqslant l_y - 1$ выполняется $\max(l_y, t+1) = l_y$, а при $t \geqslant l_y$ выполняется $\max(l_y, t+1) = t + 1$. Таким образом, точки $t = r_x$ и $t = l_y - 1$ разделяют числую прямую на три отрезка, на каждом из которых максимум и минимум раскрываются однозначно и функция $(\min(r_x, t) - l_x + 1)(r_y - \max(l_y, t+1) + 1)$ является многочленом степени не больше 2.

Чтобы сумму значений многочлена $at^2 + bt + c$ по всем $t \in [l, r]$, достаточно посчитать сумму $S_2 = \sum_{t=l}^{r} t^2$, $S_1 = \sum_{t=l}^{r} t$ и $S_0 = \sum_{t=l}^{r} 1$, тогда сумма значений многочлена будет равна $aS_2+bS_1+cS_0$. Значение S_0 равно длине отрезка $[l, r]$. Значение S_1 считается с помощью формулы суммы арифметической прогрессии а значение S_2 считается с помощью формулы суммы квадратов от 1 до n. Все это можно посчитать за $O(1)$.

При изменении a_i или b_i пересчитать нужно только значения $s_{i-1}, s_i, s_{i+1}, w_{i-1}, w_i, w_{i+1}$, поэтому запрос изменения можно обрабатывать за $O(\log M)$, где $M = 10^9 + 7$, так как для нахождения $\frac{s_i}{w_i}$ нужно найти обратное к w_i по модулю M.

Таким образом, мы научились решать задачу за $O((n + q) \log M)$.