Разбор задачи «Поедание крыс»

Рассмотрим суммы S_i, которые можно получить из первых i элементов заданного протокола. То есть $S_i = a_1 + a_2 + \cdots + a_i$.

Если существует такое i, что $S_i = k$, то максимальный отрыв будет равен k. Так как это соответствует случаю, когда все первые i действий совершил Кратос. Больше k отрыв очевидно быть не может, так как известно, что Кратос съел ровно k крыс.

В случае, если такого i не существует, то отрыв k не мог случится. Тогда ответ — $k - 1$. Он соответствует случаю, когда Атрей съел одну крысу, пока Кратос ел k.

Разбор задачи «Дневнегреческая машина»

Если $A \leqslant 1$, несложно доказать, что больше, чем $A \cdot 100$ километров Кратосу не проехать.

Если же $A > 1$, оптимальный алгоритм такой: зальем 1 литр энергии, проедем $100x$ километров, потратив x литров энергии, затем остановимся, выльем $1 - 2x$ литров, а на оставшиеся x литров энергии уедем на старт. На старте зальем оставшиеся $A-1$ литр энергии, проедем $100x$ километров, потратив x литров энергии, и зальем $1-2x$ литров, которые нами там были оставлены. Итого, у нас сейчас в баке $A - 3x$ литров энергии. Чтобы проехать как можно больше, это количество должно быть равно 1 — тогда суммарно мы проедем $(1 + x) \cdot 100$ километров. Решая несложное уравнение, получаем, что $x = (A - 1)/3$, а итоговое суммарное расстояние равно $100 \cdot (1 + (A - 1)/3)$.

Разбор задачи «Старик и шахматная доска»

У этой задачи было множество решений, и вот одно из них:

Переберем длину стороны итоговой доски — она точно не превосходит $\lceil\sqrt{n+m}\rceil$. Если перебранная длина x четная, и черных, и белых клеток в шахматной раскраске этой доски будет $x^2/2$, то есть каждое из чисел n и m должно быть не меньше этого числа. Если же x — нечетное, то черные и белые клетки делятся как $(x^2 - 1)/2$ и $(x^2 + 1)/2$, и нужно рассмотреть два случая на числа n и m.

Разбор задачи «Ящик Пандоры»

Будем считать динамику dp_i — минимальное число действий, чтобы сделать неубывающим суффикс массива от i до $n - 1$ (здесь и далее нумерация массивов с 0 до $n - 1$).

Пусть $T(l, r) = 0$, если элементы с l по r одинаковые и 1 иначе. Пойдем **с конца массива** и заметим, что

$$dp_i = dp_j + T(i, j - 1), \text{ если } a_{j-1} = a_i \text{ и } j > i.$$

Очевидно, для получения минимального количества действий, $dp_j + T(i, j - 1)$ должно быть минимально возможным. Заметим, что $dp_j + T(i, j - 1)$ минимально при минимальном dp_j, а если таких несколько, то при минимальном j.

Тогда для каждого числа a_i из входных данных будем поддерживать $p[a_i]$ — индекс минимального dp_j, такого, что $a_{j-1} = a_i$ и $a_{j-1} < a_j$. Это можно поддерживать, проходя с конца по исходному массиву в процессе подсчета динамики.

Для того, чтобы не добавлять в dp_i действие, когда мы хотим выравнять отрезок с i по j, но на нем элементы и так одинаковые, заведем массив c, при этом $c_i = 1$, если $a_i = a_{i+1}$, иначе 0. Теперь заведём массив sum, для которого sum_i — сумма чисел в массиве c на отрезке с 0 по i (массив префиксных сумм).

Теперь $q = sum[p[a[i]] - 1] - sum[i - 1]$ — количество чисел, равных a_i, на оптимальном отрезке, начинающемся с i. Тогда, если q — не равно длине этого отрезка, значит на нем существует отличное от концов отрезка число, и нужно потратить действие для уравнивания и $dp_i = dp[p[a[i]] + 1$, иначе $dp_i = dp[p[a[i]]$.

Разбор задачи «Кружок стрельбы»

Заметим, что при выстреле i-го ученика снаряд долетит до следующего ученика тогда и только тогда, когда $x_i + r_i c_i \geqslant x_{i+1}$. Значит, если для какого-то номера i выполняется неравенство $x_i > x_{i-1} + r_{i-1}c_{i-1}$, то ученику с этим номером обязательно нужно дать команду, так как снаряд до него долететь не может. Также очевидно, что до ученика с номером 1 никакой снаряд долететь не может, и ему нужно дать команду. Заметим теперь, что всем остальным ученикам давать команду не нужно, так как до каждого из них долетит снаряд.

Таким образом, чтобы получить ответ, достаточно посчитать количество чисел x_i, для которых $x_i > x_{i-1} + r_{i-1}c_{i-1}$, и увеличить на единицу. Асимптотика времени работы решения составляет $\mathcal{O}(n)$.

Разбор задачи «Древнегреческий изоморфизм»

Рассмотрим клетку (i, j) и кратчайшие расстояния от неё до углов решетки. Тогда рассмотрим расстояния до углов $(1, 1)$ и $(1, m)$. Эти расстояния равны, соответственно, $i + j$ и $i + (m - j)$. Тогда мы можем найти i как $\frac{((i+j)+(i+(m-j))-m}{2}$. И можем найти j как $\frac{((i+j)-(i+(m-j))+m}{2}$.

Таким образом, если мы знаем кратчайшие расстояния до углов решетки, то мы можем однозначно восстановить координаты клетки.

Углам решетки соответствуют вершины степени два данного графа, и если он изоморфен решетке, то таких вершин ровно четыре. Переберем 4! вариантов какая вершина, какому углу соответствует.

Тогда мы для каждой вершины можно найти по формулам выше соответствующую вершину решетки, а затем проверить, что полученная перестановка вершин действительно является изоморфизмом решетки и данного графа.

Таким образом, мы получили решение за $\mathcal{O}(4!nm)$, что есть $\mathcal{O}(nm)$.

Разбор задачи «Гонки на колесницах»

Пусть на победу колесничего было поставлено a монет. Тогда для того, чтобы Кратос гарантированно остался в выигрыше, должны выполняться два неравенства:

1. $a \cdot x + a > n \Rightarrow a \cdot (x + 1) > n \Rightarrow a > \frac{n}{x+1}$
2. $(n - a) \cdot y + (n - a) > n \Rightarrow (n - a) \cdot y > a \Rightarrow a < \frac{n \cdot y}{y+1}$

Получаем ограничение на a: $\frac{n}{x+1} < a < \frac{n \cdot y}{y+1}$.

Для рассмотрения максимального потенциального выигрыша сравним максимальный a, умноженный на $x + 1$ и максимальный $n - a$ (при минимальном a), умноженный на $y + 1$.

Разбор задачи «Игра в строки»

Пусть массив a_i — количество i-х букв алфавита, которое есть во второй строке. Зафиксируем подстроку длины k первой строки. Пусть массив b_i — количество i-х букв алфавита, которое есть в этой подстроке. Тогда надо проверить, что существует подстрока, для которой $b_i \leqslant a_i$ для всех i от 1 до 26.

Посчитаем массив a за $O(|t|)$. Также посчитаем массив b для подстроки длины k первой строки, начинающейся в первом символе. Заметим, что при переходе от одной подстроки к следующей, в массиве b изменяется не более двух элементов, а значит этот переход делается за $O(1)$.

Тогда построить массив b для всех подстрок длины k первой строки можно за $O(|s|)$. Проверить каждую из них занимает $O(26)$ времени, где 26 — размер алфавита. Тогда всё решение работает за $O(26|s| + |t|)$.