Оглавление

Сумма элементов массива

Дан массив A длины n. Найдите сумму его элементов.

Задача решается простым проходом по массиву и накоплением суммы. Изначально $s = 0$, затем для каждого элемента a_i прибавляем его к s.

Произведение элементов массива

Дан массив A длины n. Найдите произведение его элементов по модулю $10^9 + 7$.

Задача решается простым проходом по массиву: инициализируем $p = 1$, затем для каждого элемента a_i обновляем $p = (p \cdot a_i) \bmod M$, где $M = 10^9 + 7$.

Странная последовательность

Дано целое число x. Найдите длину наибольшей последовательности из квадратов натуральных чисел $1^2, 2^2, 3^2, \ldots,$ каждый элемент которой не превосходит x.

Нужно найти максимальное k, такое что $k^2 \leq x$. Это примерно $\sqrt{x}$. Перебираем $i = 1, 2, 3, \ldots,$ пока $i^2 \leq x$, и увеличиваем счётчик - ответ конечное i.

A + B

Введём новую операцию сложения: $A + B$ — это число, полученное приписыванием десятичной записи B справа к десятичной записи A (без ведущих нулей). Для заданных A и B найдите наибольшее из $A + B$ и $B + A$.

Сравниваем $A + B$ и $B + A$ как строки и выводим наибольшую.

Закраска точек

У Васи есть правильный n-угольник с вершинами $1, 2, \ldots, n$. Он выбирает число k (от 1 до $n - 1$) и закрашивает вершины по кругу с шагом k, начиная с вершины 1. Процесс идёт до тех пор, пока не встретится уже закрашенная вершина. Нужно найти такое k, при котором закрашенных вершин будет меньше всего. Если таких k несколько, выбрать наименьшее.

Если шаг k и n не взаимно просты, то закрасится не весь многоугольник, а только его часть. Количество закрашенных вершин равно $n/\gcd(n, k)$. Чем больше $\gcd(n, k)$, тем меньше вершин закрасится - поэтому нужно сделать $\gcd(n, k)$ как можно больше, то есть взять наибольший собственный делитель числа n. Искомое k и будет равно этому делителю.

Кофейные колечки

В кружке в начале дня нет кофе. При доливе кофе на стенке образуется кольцо на новом уровне, а все кольца ниже этого уровня исчезают. При выпивании кофе кольца не исчезают, но на новом уровне появляется кольцо. В конце дня нужно посчитать количество колец на кружке (кольцо на уровне 0 тоже считается).

Будем хранить в множестве все уровни, на которых есть кольца. Также поддерживаем текущий объём кофе v. При операции $+X$: увеличиваем v на X, затем удаляем из множества все уровни, меньшие v (они растворились), и добавляем уровень v (новое кольцо). При операции $-X$: уменьшаем v на X и добавляем уровень v в множество (на новом уровне появляется кольцо). В конце выводим размер множества.

Бенедикт и яблоки

У каждого из n продавцов есть две кучки яблок: a_i и b_i. Нужно купить у каждого продавца ровно одну кучку. Все купленные яблоки делятся поровну на троих, остаток забирает Бенедикт. Бенедикт хочет, чтобы его доля была положительной, и при этом общее количество купленных яблок было максимальным. Найти это максимальное количество.

Сначала для каждого продавца выберем бóльшую кучку (это даст максимальную сумму). Посчитаем эту сумму S. Если S не делится на 3 нацело, то остаток уже положителен — это и есть ответ. Если S делится на 3, то нужно уменьшить S на минимально возможную величину, чтобы остаток стал ненулевым. Для этого нужно заменить у какого-то продавца выбор с большей кучки на меньшую. Величина уменьшения равна разности $|a_i - b_i|$, причём эта разность не должна делиться на 3 (иначе остаток снова станет нулевым). Выбираем минимальную такую разность и вычитаем её из S.

Бесконечная игра

Дана строка s длины n. Фишка стоит на первой позиции. Если на поле есть ещё одна позиция с таким же символом, как на текущей, фишка может перейти на любую из них (противник выбирает переход, чтобы затянуть игру). Иначе фишка сдвигается на одну позицию вправо. Игра заканчивается, когда фишка выходит за правый край. Может ли игра продолжаться бесконечно?

Если в строке есть хотя бы два одинаковых символа, то противник может перебрасывать фишку между ними бесконечно, не давая ей двигаться вправо. Если все символы различны, то на каждой позиции есть только один вариант — двигаться вправо, и игра закончится ровно за n ходов. Таким образом, ответ YES, если все символы различны, и NO в противном случае.

Строки и столбцы

Дана квадратная таблица $n \times n$, заполненная натуральными числами от 1 до $k-1$. X — сумма минимальных элементов по столбцам, Y — сумма минимальных элементов по строкам. Найдите максимально возможное значение X/Y.

Оценка. Пусть $m = \min_{1 \le i \le n} r_i$ — наименьший из строчных минимумов, и i_0 — индекс строки, в которой этот минимум достигается, то есть $r_{i_0} = m$. Для любого столбца j по определению $c_j \le a_{i_0,j}$ (так как c_j — минимум по j-му столбцу, а $a_{i_0,j}$ — один из его элементов). Суммируя по всем j, получаем:

$$X = \sum_{j=1}^{n} c_j \le \sum_{j=1}^{n} a_{i_0,j}.$$

В строке i_0 есть хотя бы один элемент, равный m (тот, на котором достигается r_{i_0}), а все остальные элементы не превосходят $k-1$ (по условию все числа таблицы $\le k-1$). Следовательно, сумма элементов строки i_0 не больше $m + (n-1)(k-1)$. Таким образом,

$$X \le m + (n-1)(k-1). \tag{1}$$

Далее, так как $r_i \ge m$ для всех i, имеем:

$$Y = \sum_{i=1}^{n} r_i \ge n \cdot m. \tag{2}$$

Из (1) и (2) получаем:

$$\frac{X}{Y} \le \frac{m + (n-1)(k-1)}{nm} = \frac{1}{n} + \frac{(n-1)(k-1)}{nm} \le [\text{при m = 1}] \le \frac{1 + (n-1)(k-1)}{n}.$$

Пример. Сделаем так, чтобы в каждой строке была хотя бы одна 1, но при этом в одном столбце минимум был 1, а в остальных — $k-1$. Для этого составим таблицу следующим образом: перый столбец заполним единицами, а остальные ячейки числами $k-1$.

Цветы-мутанты

Даны две матрицы $n \times m$: до и после применения сыворотки. Гипотеза: существует функция f, такая что для каждой клетки $b_{i,j} = f(a_{i,j})$. Проверить, верна ли гипотеза.

Проверяем, что одинаковым значениям в первой матрице соответствуют одинаковые значения во второй матрице. Для этого заводим массив $mapping$ (от 1 до 10^5), изначально заполненный -1. Проходим по всем клеткам: для текущей пары (x, y):

- Если $mapping[x] = -1$, запоминаем $mapping[x] = y$.
- Иначе если $mapping[x] \neq y$, то гипотеза неверна.

Также заметим, что x не может равняться y (каждый цветок должен измениться), поэтому дополнительно проверяем $x \neq y$. Если все проверки пройдены, ответ YES, иначе NO.

Загадай желание

За круглым столом сидят n человек ($n \geq 3$), у каждого имя — буква. Можно переименовывать людей (менять букву) и пересаживать (переставлять буквы в строке). Нужно добиться, чтобы у каждого человека оба соседа имели одинаковые имена. То есть для циклической строки: $s_{i-1} = s_{i+1}$ для всех i. Найти минимальное количество переименований (перестановки бесплатны).

Если n нечётное, то из условия $s_{i-1} = s_{i+1}$ следует, что все символы равны, потому что двигаясь по кругу, получаем $s_1 = s_3 = s_5 = \cdots = s_n = s_2 = s_4 = \cdots = s_{n-1}$. Значит, вся строка состоит из одной буквы. Минимальное количество переименований — n минус максимальное количество одинаковых букв в исходной строке (остальные переименовываем в эту букву).

Если n чётное, то строка должна чередовать две буквы: $ababab$... (в цикле). Буквы могут совпадать (тогда это частный случай при нечетном n). Нужно выбрать две буквы a и b (возможно, $a = b$) так, чтобы минимизировать количество несовпадений. Для фиксированных a и b стоимость равна

$n -$ (количество позиций a на нечётных местах) $-$ (количество позиций b на чётных местах),

но так как мы можем переставлять людей, мы можем разместить всех с буквой a на нечётные места, а с буквой b — на чётные. Поэтому стоимость равна $n - \min(cnt[a], n/2) - \min(cnt[b], n/2)$, где cnt — частоты букв в исходной строке. Перебираем все пары (a, b) и находим минимум.

Лягушка и строка

Лягушка начинает на первом символе строки S длины n и хочет попасть на последний. За один прыжок можно переместиться вправо не более чем на k позиций. Нельзя прыгать между символами разного типа (строчные буквы, заглавные буквы, цифры — три типа). Найти минимальное количество прыжков или -1, если невозможно.

Если тип первого символа не совпадает с типом последнего, то ответ -1 (иначе последний символ недостижим, так как типы не меняются). Рассмотрим только позиции с нужным типом (таким же, как у первого символа). Построим по ним граф: из позиции i можно прыгнуть в позицию j, если $j > i$ и $j - i \leq k$ (по индексам в исходной строке, но с учётом только нужных позиций). Тогда задача сводится к поиску кратчайшего пути в этом графе. Так как граф является "интервальным"(из каждой позиции можно добраться до всех следующих в пределах k), можно применить BFS с оптимизацией: поддерживаем указатель на первую непосещённую позицию и при обработке вершины i помечаем все позиции до $far[i]$ как достижимые за один прыжок. $far[i]$ — самая дальняя позиция с нужным типом, до которой можно допрыгнуть из i.