Задача A. Арифметические операции и двоичная последовательность

Так как $a+1 > a \cdot 1$, $a+0 \geqslant a \cdot 0$ для всех неотрицательных a, то в случае, если в варианте расстановки, реализующем минимум, есть сложение, замена его умножением не увеличит результат. Аналогично если в варианте, реализующем максимум, умножение заменить сложением результат по крайней мере не уменьшится. Поэтому для получения максимума все числа во входе надо сложить, а для получения минимума — перемножить.

Задача B. Балкон

Посчитаем для каждой задачи набор решивших её команд. Если для каких-то двух задач эти два набора неразличимы, то различить цвета шариков не получится. Если же все наборы команд различны, то зритель может сопоставить каждому цвету шарика набор команд, его получивших, и восстановить соответствие, используя табличку.

Испольуя структуру данных set, можно добиться сложности $O(p \cdot t \cdot \log p)$ по времени, а используя битсеты, улучшить константу. Впрочем, в ограничениях задачи проходили и решения за $O(p^2 t)$

Задача C. Варианты легенды

Рассмотим количество зёрен на клетке $a_{i,j}$ и на соседних клетках. Количество зёрен на соседней сверху по вертикали клетке в 2^n раз больше, количество зёрен в соседней справа по горизонтали клетке вдвое больше, количество зёрен в соседней по диагонали клетке больше в 2^{n+1} раз в случае, если направление считается вдоль диагонали 1 (требуется «пройти» n+1 шаг), и в 2^{n-1} раз в случае, если направление считается вдоль диагонали 2.

Поэтому в каждом случае ответом будет сумма той или иной геометрической прогрессии со знаменателем, равным степени двойки.

На первой горизонтали будет $1+2+\ldots+2^{n-1} = 2^n - 1$ зёрен, на i-й тем самым — $2^{(i-1)n} \cdot (2^n - 1)$ зёрен.

На самой левой вертикали будет $1 + 2^n + (2^n)^2 + \ldots + (2^n)^{n-1} = (2^{n^2} - 1)/(2^n - 1)$ зёрен, соответственно, на i-й слева вертикали будет $2^{i-1} \cdot (2^{n^2} - 1)/(2^n - 1)$ зёрен.

На диагонали от $(1,1)$ до (n,n) будет $1+2^{n+1}+(2^{n+1})^2+\ldots+(2^{n+1})^{n-1} = (2^{n(n+1)}-1)/(2^{n+1}-1)$ зёрен.

На диагонали от $(1,n)$ до $(n,1)$ будет $2^{n-1} \cdot (1+2^{n-1}+\ldots+(2^{n-1})^{n-1}) = 2^{n-1} \cdot (2^{n(n-1)}-1)/(2^{n-1}-1)$ зёрен.

Для вычисления соответствующих остатков нужно реализовать быстрое возведение в степень (для подзадач 3 и 4), а также деление по простому модулю (например, через Малую теорему Ферма) для поздадачи 4.

Задача D. Интересные тройки

Заметим, что $a + b = a \text{ xor } b +$ переносы. Значит, переносы $= 0$. То есть в числах a, b нет общих единичных битов. Из третьего условия следует, что нет общих и нулевых битов. То есть там, где в a 0, в b 1, и наоборот. Тогда $a_k = 1111...111_2 = 2^b - 1$. Решение: перебираем j, поддерживая слева от j массив cnt – количество каждого числа. Тогда тройка выглядит как: $(2^b - 1) \text{ xor } a_j, a_j, 2^b - 1$. Прибавляем к ответу $cnt[(2^b - 1) \text{ xor } a_j] * c(j+1)$, где $c(j+1)$ – количество чисел $2^b - 1$ справа от позиции j. Такое количество можно поддерживать в переменной.

Задача E. Интересное свойство

Если число записывается в виде однозначного, то разность между ним и суммой цифр равна нулю, который простым не является. Значит, надо рассматривать основания, не большие x.

Пусть у нас $x = a_i b^i + a_{i-1} b^{i-1} + \ldots + a_1 b + a_0$, тогда разность x и суммы цифр равна сумме $a_k(b^k - 1)$, а так как $b^k - 1 = (b-1) \cdot (b^{k-1} + \ldots + b + 1)$, то разность делится на $b - 1$. Поэтому $b - 1$ или равно 1, или $b - 1$ — простое. Далее заметим, что второй множитель равен $a_i \cdot b^{i-1} + \ldots$, поэтому для того, чтобы он был равен 1, требуется, чтобы число было двузначным с первой цифрой 1. При $b - 1 = 1$ второй сомножитель должен быть простым.

Так что нам достаточно явно проверить разность при $b = 2$ на простоту, а для всех остальных случаев задача сводится к нахождению количества простых чисел между $x/2+1$ и x включительно.

Если найти простые числа до 10^8 с помощью решета Эратосфена (например, над `vector <bool>`, а затем построить таблицу частичных сумм $ps(i)$ — количество простых чисел с начала и до i, то обе проверки (и разности x с суммой его бит на простоту, и количества простых чисел в нужном диапазное) можно выполнить за $O(1)$.