Эрен и подвал

Автор задачи и разработчик: Николай Ведерников

Для решения **первой подзадачи** было достаточно просимулировать описанный в условии процесс. Будем перебирать все возможные значения второго счетчика, начиная с исходного b, каждый раз проверяя, существует ли у a и текущего значения b общий делитель. Проверять существование общего делителя в первой подзадаче можно было за линейное время, просто перебирая все числа от 2 до a. Такое решение работает за время $\mathcal{O}(ta \cdot ans)$, где ans — величина ответа.

Чтобы уточнить эту оценку, докажем следующий факт: для того, чтобы открыть замок, потребуется строго меньше a секунд. Действительно, среди чисел от b до $b + a - 1$ всегда есть ровно одно число, которое делится на a. Таким образом, менее чем через a секунд наступит состояние, в котором оба значения, на которые указывают механизмы, будут иметь общий делитель, равный $a > 1$. Следовательно, время работы решения в первой подгруппе равно $\mathcal{O}(ta^2)$.

Чтобы решить **вторую подзадачу**, заметим, что на самом деле можно быстрее проверять, подходят ли конкретные a и b под условие, гарантирующее открытие замка. У чисел a и b существует общий делитель, больший единицы, тогда и только тогда, когда их *наибольший* общий делитель больше 1. Найти наибольший общий делитель двух чисел можно алгоритмом Евклида за время $\mathcal{O}(\min(a, b))$. Получаем решение, которое работает за $\mathcal{O}(ta \log(a + b))$.

По ограничениям **третьей подзадачи** a — простое число, а тогда его единственными натуральными делителями являются 1 и a. Таким образом, если у a и b есть общий делитель, больший единицы, то этот общий делитель равен a. В итоге задача сводится к тому, чтобы найти наименьшее число, большее либо равное b, которое при этом делится на a. Такое число можно найти за время $\mathcal{O}(1)$ по формуле $(-b) \bmod a$. Действительно, для числа, делящегося на a, эта формула даст результат 0, а для числа с большим остатком даст остаток, дополняющий его до нулевого.

При взятии остатка в большинстве языков программирования при этом следует не забывать, что взятие отрицательного числа по модулю дает отрицательное число в результате, поэтому полная формула для ответа будет выглядеть как $((-b) \bmod a + a) \bmod a$. Либо, альтернативно, можно проверить число $(-b) \bmod a$ на отрицательность, и в таком случае прибавить a.

Полное решение основано на идеях третьей подзадачи. Поскольку число a не меняется, то общий делитель a и итогового значения b — это, в частности, один из делителей числа a. Заметим, что всего различных натуральных делителей у числа a не больше $2\sqrt{a}$. Действительно, рассмотрим отдельно те делители, которые не превосходят $\sqrt{a}$, и те, которые больше него. Первые принимают значения от 1 до $\sqrt{a}$, а значит их количество тоже не превосходит $\sqrt{a}$. А на каждый делитель $x > \sqrt{a}$ существует свой «парный» делитель $\frac{a}{x} < \sqrt{a}$, поэтому их не больше, чем первых.

Таким образом, в каждом тестовом случае достаточно за время $\mathcal{O}(\sqrt{n})$ перебрать все делители числа a, и для каждого из них проверить, какой будет ответ, если добиваться, чтобы итоговое b делилось именно на этот делитель a. По всем полученным ответам достаточно выбрать максимум.

А найти минимальное подходящее для второго механизма число, опять же, можно за время $\mathcal{O}(1)$: если мы зафиксировали делитель числа a, равный x, то минимальный подходящий ответ равен $(-b) \bmod x$. Конечное решение, перебирающее все делители a в каждом тестовом случае, работает за время $\mathcal{O}(t\sqrt{a})$.