
Быстрый перевод

Базовое решение — пройти по всем i от 59 до 0 в убывающем порядке, пытаясь перевести 2^i . Поскольку по условию $x \leq 10^{18} < 2^{60}$, то этот алгоритм будет просто по очереди заменять в двоичной записи числа x единицы на нули в порядке убывания степени двойки, а значит после его завершения x станет равен нулю.

Поскольку ограничения задачи не позволяли для любого x делать 60 запросов, требовалось улучшить это значение. В частности, можно было бинарным поиском найти максимальную степень двойки, не превосходящую x .

Для этого сделаем двоичный поиск по i в интервале $[0, 60)$. Если на запрос перевода 2^m мы получаем ответ **accepted**, переместим указатель на левую границу ($l = m$), иначе — на правую границу ($r = m$). Поскольку в процессе выполнения таких запросов x мог изменяться, не факт, что мы найдем именно максимальную степень двойки, не превышающую его, но можно доказать следующее **утверждение**: если после такого двоичного поиска выполнено $x \leq 2^t$, то так же выполнено $l \leq t$.

Чтобы доказать это утверждение, достаточно посмотреть на последний раз, когда мы получили **accepted**. Если после этого были получены только **rejected**, это значит, что ни для какого $m > l$, x не превышает 2^m . Это ровно то, что мы хотели доказать.

Осталось только повторить базовое решение, только начиная не с $i = 59$, а с $i = l$. Так как x не увеличился, l не превышает двоичного логарифма исходного x , а весь бинарный поиск занял не более 6 операций, так как $60 < 2^6$. Следовательно, такой алгоритм переводит все деньги, даже имея в запасе 4 операции по сравнению с ограничениями в условии.