

Задача из Копейска

Рассмотрим два целых положительных числа x и y . Пусть они раскладываются на простые множители следующим образом: $x = p_1^{k_1} \cdot p_2^{k_2} \cdot \dots \cdot p_c^{k_c}$ и $y = p_1^{m_1} \cdot p_2^{m_2} \cdot \dots \cdot p_c^{m_c}$. Тогда, чтобы x было кратно y , необходимо и достаточно, чтобы для каждого простого множителя в разложении y его степень была меньше или равна степени этого множителя в разложении x . Формально, $\forall i \in [1, c] \ k_i \geq m_i$.

Значение $f(x)$ из условия задачи должно быть кратно каждому числу от 1 до x включительно по определению наименьшего общего кратного. Получается, если в разложении на простые множители $f(x) = p_1^{k_1} \cdot p_2^{k_2} \cdot \dots \cdot p_c^{k_c}$ и $y = p_1^{m_1} \cdot p_2^{m_2} \cdot \dots \cdot p_c^{m_c}$, то $\forall y \in [1, x] \ \forall i \in [1, c] \ k_i \geq m_i$, иными словами, степень каждого простого числа в разложении $f(x)$ на простые множители должна быть больше или равна максимальной степени этого простого числа, встречающейся в разложениях чисел от 1 до x (то есть префиксному максимуму степени этого простого числа в разложении чисел от 1 до x). Поскольку $f(x)$ — это **наименьшее** общее кратное, то мы должны взять минимально возможное k_i для каждого i , поэтому $\forall i \in [1, c] \ k_i = m_i$.

Теперь рассмотрим ряд $f(1), f(2), f(3), \dots, f(x), \dots$. Заметим, что значения в нём не убывают, так как, как мы уже выяснили, $f(x)$ зависит от префиксных максимумов степеней простых чисел в разложении, а префиксные максимумы не убывают.

Для начала решим получившуюся подзадачу нашей задачи: есть неубывающая последовательность чисел, нужно быстро находить количество различных чисел на отрезке $[l, r]$. Для этого нам нужно найти все индексы i такие, что $f(i) > f(i-1)$. Сформируем массив a такой, что $a[i] = 1$, если $i = 1$ или $f(i) > f(i-1)$, иначе $a[i] = 0$ (о том, как сформировать такой массив, будет написано ниже). Иными словами, единицы в массиве a будут в тех позициях, где $f(i)$ меняет своё значение по сравнению с предыдущим. Тогда количество различных чисел на отрезке $[l, r]$ будет равно количеству единиц на отрезке $[l, r]$ в массиве a (возможно, $+1$). Почему так? Пройдёмся слева направо по отрезку $[l, r]$ в массиве a . Если $a[i] = 1$, то это новое значение $f(i)$, добавим 1 к ответу. Иначе $a[i] = 0$, значит, такое значение $f(i)$ уже встречалось, и в массиве a где-то ранее была единичка, которую мы уже прибавили к ответу, поэтому ничего делать не нужно. Исключение из этого правила — это позиция $a[l]$, в которой может быть 0, для которого мы не прибавили к ответу единичку (так как она была левее отрезка $[l, r]$). Поэтому если $a[l] = 0$, то к количеству различных чисел на отрезке надо прибавить 1. Количество единичек в массиве a на отрезке можно посчитать при помощи префиксных сумм за $O(1)$ с предподсчётом за $O(\max r)$ ($\max r$ в задаче равно 10^7). В итоге мы научились быстро (за $O(1)$) отвечать на запрос количества различных чисел на отрезке $[l, r]$, имея массив a .

Теперь поговорим о том, как получить вышеупомянутый массив a . Для этого можно воспользоваться решето Эратосфена. Сначала присвоим $a[1] = 1$. Затем, когда мы в процессе выполнения алгоритма нашли очередное простое число p , то во внутреннем цикле, когда мы проходимся по всем числам, кратным p , чтобы пометить их как составные, нужно отдельно для каждого числа z , равного p^k , где k — целое положительное число и $k \geq 1$, присвоить $a[z] = 1$. Таким образом, $a[i]$ станет равно 1 в тех индексах i , где увеличивается префиксный максимум степени какого-либо простого числа из разложения на простые множители, а значит, как мы уже выяснили, увеличивается и значение функции f — такой массив a нам и нужен! Это всё нужно проделать один раз до ответа на запросы. После этого мы получим массив a , по которому построим массив префиксных сумм и сможем отвечать на запросы за $O(1)$.

Итого наш алгоритм выполняет решето Эратосфена за $O(\max r \cdot \log \log \max r)$, формирует массив префиксных сумм за $O(\max r)$, а затем отвечает на t запросов по $O(1)$ на каждый. Итоговая асимптотика получается $O(\max r \cdot \log \log \max r + t)$.