

Теория чисел. Разбор задач

1 Алгоритм Евклида

Задача 1. Научитесь делать алгоритм Евклида для длинных чисел за $O(n^2)$, где n — длина числа.

Решение. Будем действовать таким образом:

1) Если оба числа делятся на 2, то $\gcd(a, b) = 2 \cdot \gcd(\frac{a}{2}, \frac{b}{2})$.

2) Если ровно 1 число делится на 2, будем считать, что это a . Тогда $\gcd(a, b) = \gcd(\frac{a}{2}, b)$.

3) В противном случае делаем через вычитание (при $a \geq b$ $\gcd(a, b) = \gcd(b, a - b)$).

Замечаем, что после действий 1) и 2) хотя бы одно из чисел уменьшается в 2 раза. Так же замечаем, что после действия 3) одно из чисел становится чётным, значит далее мы совершим действие 2). А значит за каждые 2 действия хотя бы одно из чисел уменьшается в 2 раза, таким образом мы найдём \gcd 2-х чисел за $O(n)$ действий, каждое из которых мы совершаем за $O(n)$.

Задача 2. Оцените время нахождения НОД набора из n чисел, не больших чем C .

Решение.

Лемма. Пусть $a \geq b$. Тогда $a \% b \leq \frac{a}{2}$.

Доказательство. Пусть $b \geq \frac{a}{2}$. Тогда очевидно, что $a \% b = a - b \leq \frac{a}{2}$. Иначе, если $b < \frac{a}{2}$, то $a \% b < b < \frac{a}{2}$.

Покажем, что за три шага алгоритма Евклида:

- Наибольшее из чисел уменьшится хотя бы в 2 раза.
- Наибольшее из чисел станет не больше, чем то, что было минимальным.

Пусть находили $\gcd(a, b)$.

Случай 1. $a \geq b$. Тогда на следующем шаге будем находить $\gcd(b, a \% b)$, а потом $\gcd(a \% b, b \% (a \% b))$, притом очевидно, что $b \geq a \% b$. Из нашей леммы следует, что $a \% b \leq \frac{a}{2}$, а также $b \% (a \% b) \leq \frac{b}{2}$. То есть оба числа уменьшились хотя бы в 2 раза после двух шагов. Второе свойство тривиально и не требует объяснений.

Случай 2. $a < b$. Тогда после одного шага перейдём к случаю 1, и за следующие 2 шага сделаем, что требуется.

Вычисление НОД выглядит так:

$$\gcd(a_1, a_2, \dots, a_n) = \gcd(\gcd(\dots \gcd(a_1, a_2), a_2) \dots, a_n)$$

Тогда в процессе вычисления НОД у нас будет $n - 1$ раз находится \gcd от НОД первых i чисел и $(i + 1)$ -го числа, и после каждого такого дописывания $(i + 1)$ -го числа в результате не более трёх шагов алгоритма оба числа станут меньше или равны НОД первых i чисел — это шаги первого типа.

У нас останутся какие-то другие шаги, из которых каждые 3 подряд идущих уменьшают максимальное из чисел хотя бы в 2 раза. При этом НОД всегда не больше, чем максимальное из чисел, поэтому таких шагов суммарно $O(\log C)$ — это шаги второго типа.

Шагов первого типа $O(n)$, а второго типа $O(\log C)$, то есть всего шагов алгоритма Евклида будет сделано $O(n + \log C)$.

Задача 3. Пусть a, b, c — целые числа. Рассмотрим уравнение $ax + by = c$ относительно целых x, y .

а) Покажите, что, если c не делится на $\gcd(a, b)$, решений нет.

Решение. Так как $\gcd(a, b) | a$ и $\gcd(a, b) | b$, то $\gcd(a, b) | ax + by$. Значит, если решение существует, то $\gcd(a, b) | c$.

б) Покажите, что при $c = \gcd(a, b)$, решение есть.

Решение. См. расширенный алгоритм Евклида.

в) Покажите, что, решение есть тогда и только тогда, когда c делится на $\gcd(a, b)$.

Решение. Если c не делится на $\gcd(a, b)$ то по пункту а) решений нет. Иначе $c = d \cdot \gcd(a, b)$. У нас есть решение уравнения $ax + by = \gcd(a, b)$. Домножим x и y на d и получим решение исходного уравнения.

г) Покажите, что, если существует хотя бы одно решение, существует бесконечно много решений. Опишите их все.

Решение. Заметим, что если существует решение $ax_0 + by_0 = c$, то возьмём $x_1 = x_0 + b/\gcd(a, b)$, $y_1 = y_0 - a/\gcd(a, b)$. Эта пара значений так же удовлетворяет уравнению. Так мы можем прибавить $b/\gcd(a, b)$ к x и вычесть $a/\gcd(a, b)$ из y бесконечное число раз. Тогда число решений бесконечно. Заметим, что если есть 2 решения x_1, y_1 и x_2, y_2 , то $a \cdot (x_1 - x_2) + b \cdot (y_1 - y_2) = 0$. Тогда $a \cdot (x_1 - x_2) = b \cdot (y_2 - y_1)$. Тогда существует число d что $(b/\gcd(a, b)) \cdot d = x_1 - x_2$ и $(a/\gcd(a, b)) \cdot d = y_2 - y_1$. Таким образом мы можем описать все решения.

Определение. Числа a, b называются взаимно обратными по модулю m , если $a \cdot b \equiv 1 \pmod{m}$.

Задача 4. Дано число a . Надо найти за $O(\log m)$ обратное ему по не обязательно простому модулю m или определить, что такого не существует.

Решение. Пусть число x обратно числу a . Тогда $m | (a \cdot x - 1)$. Тогда существует число y что $a \cdot x - 1 = m \cdot y$. Тогда мы свели задачу к уравнению $a \cdot x - m \cdot y = 1$. В предыдущей задаче мы научились его решать.

2 Делители

Задача 5. Докажите, что $\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n} = O(\log n)$.

Решение.

$$\frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n} < \frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n/2} + (n/2) \cdot \frac{1}{n/2} < \frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n/4} + (n/4) \cdot \frac{1}{n/4} + 1 < \dots < \log_2 n$$

Задача 6. Пусть $\tau(n)$ — количество натуральных делителей n . Докажите, что $\sum_{i=1}^n \tau(i) = O(n \log n)$.

Решение. Замечаем, что 1 - делитель у каждого первого числа, 2 - делитель у каждого второго числа и т. д. и n - делитель у каждого n -го числа. Тогда

$$\sum_{i=1}^n \tau(i) = \frac{n}{1} + \frac{n}{2} + \dots + \frac{n}{n} = O(n \log n)$$

Определение. Функция f называется мультипликативной, если $f(n \cdot m) = f(n) \cdot f(m)$ для любых взаимно простых чисел n, m .

Задача 7. Пусть f, g — мультипликативные функции. Докажите, что функция $h(n) = f(n) \cdot g(n)$ мультипликативна.

Решение. Возьмём любые 2 взаимно простые числа n и m . Тогда

$$h(n \cdot m) = f(n \cdot m) \cdot g(n \cdot m) = f(n) \cdot f(m) \cdot g(n) \cdot g(m) = f(n) \cdot g(n) \cdot f(m) \cdot g(m) = h(n) \cdot h(m)$$

Задача 8. Пусть f — мультипликативная функция. Докажите, что для любого k функция $g(n) = \sum_{d|n} d^k \cdot f(d)$ мультипликативна. ($d|n$ — d делит n)

Решение. Возьмём любые 2 взаимно простые числа n и m . Так как $\gcd(n, m) = 1$, то

$$\sum_{d|(nm)} d^k \cdot f(d) = \sum_{d_1|n} \sum_{d_2|m} (d_1 \cdot d_2)^k \cdot f(d_1 \cdot d_2) = \sum_{d_1|n} \sum_{d_2|m} d_1^k \cdot d_2^k \cdot f(d_1) \cdot f(d_2) = \left(\sum_{d_1|n} d_1^k \cdot f(d_1) \right) \cdot \left(\sum_{d_2|m} d_2^k \cdot f(d_2) \right)$$

Задача 9. Докажите, что следующие функции мультипликативны:

а) $\tau(n)$ — количество натуральных делителей n .

Решение. Воспользуемся предыдущей задачей. Возьмём $k = 0$ и $f(d) = 1$ для всех d . Получим в точности то, что требуется.

б) $\sigma(n)$ — сумма натуральных делителей n .

Решение. Воспользуемся предыдущей задачей. Возьмём $k = 1$ и $f(d) = 1$ для всех d . Получим в точности то, что требуется.

3 Другие интересные задачи теории чисел

Задача 10. За $O(n)$ для всех чисел от 1 до n найдите:

Примечание. Используется реализация решета Эратосфена за $O(n)$, которая позволяет для каждого числа найти его наименьший простой делитель.

а) В какой степени минимальный простой делитель входит в его разложение.

Решение. При подсчёте решета Эратосфена добавляем вторую динамику — степень минимального простого делителя.

б) Количество его простых делителей.

Решение. При подсчёте решета Эратосфена добавляем вторую динамику — кол-во простых делителей.

в) Количество его делителей.

Решение. Будем использовать пункт а). Пусть у нас число x , мин. простой делитель — p и он входит в степени q . Тогда воспользуемся задачей 9) и поймём, что количество делителей x равно произведению количества делителей p^q и количества делителей $x/(p^q)$. Так мы постепенно насчитаем это динамику.

г) Сумму его делителей.

Решение. Аналогично пункту в)

д) Функцию Эйлера от него.

Решение. Аналогично пункту в)

Задача 11. Научитесь вычислять $a \cdot b$ для натуральных a, b , используя только сложение, деление на 2 (в том числе с остатком), а также проверку на равенство 1 за $O(\log a)$ операций сложения.

Решение: Покажем как будет работать $mult(a, b)$

1) Если остаток от деления b на 2 равен нулю, то $mult(a, b) = mult(a + a, b/2)$

2) Если остаток от деления b на 2 равен единице, то $mult(a, b) = a + mult(a + a, b/2)$

Так за каждый шаг b уменьшается в 2 раза. То есть в конце мы получим ответ за $O(\log b)$ действий.

Задача 12. (Дискретное логарифмирование) $a^x \equiv b \pmod{m}$, a и m взаимнопросты. Найти решение или определить, что его не существует, за время $O(\sqrt{m} \log m)$.

Решение. Представим x в виде $ky - r$ для $k = \lfloor \sqrt{m} \rfloor$. Тогда $a^{ky-r} \equiv b \pmod{m}$. Тогда $a^{ky} \equiv b \cdot a^r \pmod{m}$. Так как $k = \lfloor \sqrt{m} \rfloor$, то $y \leq \lfloor \sqrt{m} \rfloor$ и $r \leq \lfloor \sqrt{m} \rfloor$. Насчитаем все возможные a^{ky} и $b \cdot a^r$, а потом проверим есть ли совпадения.

Задача 13. Найти сумму gcd по всем подотрезкам массива натуральных чисел, не больших C , за $O(n \log C)$.

Решение.

Лемма. Если при добавлении числа в множество НОД всех чисел в нём изменяется, то НОД уменьшается хотя бы в 2 раза.

Доказательство. Новый НОД — делитель старого.

Будем идти слева-направо, фиксируя правую границу отрезка r . Тогда для заданного r , если будем двигать $l = r \dots 1$, получим $O(\log C)$ различных значений НОД. Будем поддерживать список отрезков левых границ с равным НОД — элементов в нём будет $O(\log C)$. Имея этот список, легко посчитать сумму НОД для отрезков с таким правым концом. Осталось научиться переходить от правой границы r к $r + 1$. Пусть $(r + 1)$ -е число равно x . Тогда это делается так:

1. Для отрезка левых границ с НОД, равным g делаем: $g = gcd(g, x)$.
2. В список добавляем отрезок левых границ $[r + 1; r + 1]$ со значением НОД x .
3. Если какие-то соседние отрезки левых границ в списке стали иметь равный НОД, нужно их объединить в один. Это обеспечит нам то, что в любой момент в списке $O(\log C)$ элементов.

Задача 14. Найти сумму gcd по всем непустым подмножествам массива из n натуральных чисел, не больших C , за $O(n + C \log C)$. Ответ найдите по модулю $10^9 + 7$.

Решение. Заведём $cnt[x]$ — количество чисел, равных x . Посчитаем $d[x]$ — количество чисел, делящихся на x . Он вычисляется за $O(C \log C)$ так:

$$d[x] = \sum_{i=1}^{\lfloor C/x \rfloor} cnt[i \cdot x].$$

Далее, посчитаем $sets[x]$ — количество множеств с НОД, равным x . Он вычисляется в обратном порядке так:

$$sets[x] = 2^{d[x]} - 1 - \sum_{i=2}^{\lfloor C/x \rfloor} sets[i \cdot x].$$

Ответ — это просто $\sum_{i=1}^C i \cdot sets[i]$.

Задача 15. Дан массив из n натуральных чисел, не больших C . Выпишем gcd по всем непустым подмножествам этого массива. Найдите медиану выписанных чисел за $O(n \cdot C \log C)$.

Решение. Аналогично предыдущей задаче, но нужно воспользоваться длинной арифметикой: представления количеств подмножеств имеют длину $O(n)$.

Задача 16. Назовём натуральное число кубастым, если его можно представить в виде $a^3 \cdot b$ для каких-то натуральных $a > 1, b \geq 1$. Найти количество кубастых чисел, не больших n . n до 10^{18} .

Решение. Будем перебирать a ($a \leq n^{\frac{1}{3}}$), свободные от квадратов, и делать вычитания аналогично задаче 14.