Введение: наибольший общий делитель

Александр Куликов

Онлайн-курс «Алгоритмы: теория и практика. Методы»
http://stepic.org/217

Наибольший общий делитель

Определение

Наибольшим общим делителем (НОД) неотрицательных целых чисел a и b называется наибольшее целое d, которое делит и a, и b.

Наибольший общий делитель

Определение

Наибольшим общим делителем (НОД) неотрицательных целых чисел a и b называется наибольшее целое d, которое делит и a, и b.

Вычисление НОД

Вход: целые числа $a, b \geq 0$.

Выход: НОД(a, b).

Наивный алгоритм

Функция $\textsc{NaiveGCD}(a, b)$

$gcd \leftarrow 1$
для d от 2 до $\max(a, b)$:
  если $d \mid a$ и $d \mid b$:
    $gcd \leftarrow d$
вернуть gcd

Наивный алгоритм

Функция NaiveGCD(a, b)

$gcd \leftarrow 1$
для d от 2 до $\max(a, b)$:
  если $d \mid a$ и $d \mid b$:
    $gcd \leftarrow d$
вернуть gcd

- Время работы: примерно $\max\{a, b\}$.
- Работает очень медленно уже даже на числах, состоящих из десяти знаков.

Лемма

Лемма

Пусть $a \geq b > 0$ и r — остаток от деления a на b. Тогда

$$\text{НОД}(a, b) = \text{НОД}(r, b).$$

Лемма

Лемма

Пусть $a \geq b > 0$ и r — остаток от деления a на b. Тогда

$$\text{НОД}(a, b) = \text{НОД}(r, b).$$

Доказательство

Достаточно доказать, что $\text{НОД}(a, b) = \text{НОД}(a - b, b)$.

Лемма

Лемма

Пусть $a \geq b > 0$ и r — остаток от деления a на b. Тогда

$$\text{НОД}(a, b) = \text{НОД}(r, b).$$

Доказательство

Достаточно доказать, что $\text{НОД}(a, b) = \text{НОД}(a - b, b)$.

- $\text{НОД}(a, b) \leq \text{НОД}(a - b, b)$: если d делит a и b, то делит и $a - b$.

Лемма

Лемма

Пусть $a \geq b > 0$ и r — остаток от деления a на b. Тогда

$$\text{НОД}(a, b) = \text{НОД}(r, b).$$

Доказательство

Достаточно доказать, что $\text{НОД}(a, b) = \text{НОД}(a - b, b)$.

- $\text{НОД}(a, b) \leq \text{НОД}(a - b, b)$: если d делит a и b, то делит и $a - b$.
- $\text{НОД}(a, b) \geq \text{НОД}(a - b, b)$: если d делит $a - b$ и b, то делит и $a = (a - b) + b$. $\square$

Алгоритм Евклида

Функция $\textsc{EuclidGCD}(a, b)$

```
если a = 0:
  вернуть b
если b = 0:
  вернуть a
если a ≥ b:
  вернуть EuclidGCD(a mod b, b)
если b ≥ a:
  вернуть EuclidGCD(a, b mod a)
```

Пример

$$\text{НОД}(3918848, 1653264)$$

Пример

$$\begin{aligned} &\text{НОД}(3918848, 1653264)\\ =&\text{НОД}(612320, 1653264) \end{aligned}$$

Пример

$$\begin{aligned} &\text{НОД}(3918848, 1653264)\\ =&\text{НОД}(612320, 1653264)\\ =&\text{НОД}(612320, 428624)\end{aligned}$$

Пример

$$
\begin{aligned}
&\text{НОД}(3918848, 1653264)\\
=&\text{НОД}(612320, 1653264)\\
=&\text{НОД}(612320, 428624)\\
=&\text{НОД}(183696, 428624)
\end{aligned}
$$

Пример

$$
\begin{aligned}
&\text{НОД}(3918848, 1653264)\\
=&\text{НОД}(612320, 1653264)\\
=&\text{НОД}(612320, 428624)\\
=&\text{НОД}(183696, 428624)\\
=&\text{НОД}(183696, 61232)
\end{aligned}
$$

Пример

$$
\begin{aligned}
&\text{НОД}(3918848, 1653264)\\
=&\text{НОД}(612320, 1653264)\\
=&\text{НОД}(612320, 428624)\\
=&\text{НОД}(183696, 428624)\\
=&\text{НОД}(183696, 61232)\\
=&\text{НОД}(0, 61232)
\end{aligned}
$$

Пример

$$
\begin{aligned}
&\text{НОД}(3918848, 1653264)\\
=&\text{НОД}(612320, 1653264)\\
=&\text{НОД}(612320, 428624)\\
=&\text{НОД}(183696, 428624)\\
=&\text{НОД}(183696, 61232)\\
=&\text{НОД}(0, 61232)\\
=&61232.
\end{aligned}
$$

Лемма

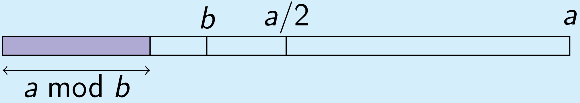
Если $a \geq b > 0$, то $a \bmod b < a/2$.

Лемма

Если $a \geq b > 0$, то $a \bmod b < a/2$.

Доказательство

- Если $b \leq a/2$, то $a \bmod b < b \leq a/2$.

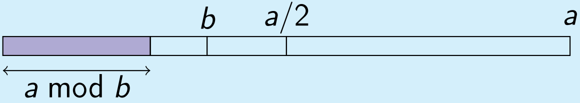


Лемма

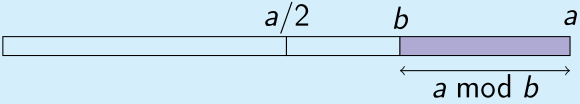
Если $a \geq b > 0$, то $a \bmod b < a/2$.

Доказательство

- Если $b \leq a/2$, то $a \bmod b < b \leq a/2$.



- Если же $b > a/2$, то $a \bmod b = a - b < a/2$.



Время работы

- Каждый шаг уменьшает одно из чисел хотя бы вдвое.
- Количество шагов: не более $\log_2 a + \log_2 b$.
- Каждый шаг — это деление.
- Вычисление НОД двух чисел из ста десятичных знаков производится за примерно 600 шагов.
- Гораздо быстрее наивного алгоритма.

Заключение

- Наивный алгоритм опять слишком медленный.
- Правильный алгоритм гораздо более быстр.
- Нахождение правильного алгоритма требует знания некоторых свойств задачи.