

# Алгоритм Евклида

Самостоятельно - алгоритм Евклида и метод нахождения НОД в «лоб».

Сравним их для  $a=1000000$  и  $b=2$

Операции	Первый вариант	Второй вариант
Операции присваивания	500 000	4
Операция сложения	499 999	0
Операции умножения	0	1
Операция сравнения	999 999	1

```

procedure gcd_simple (a,b:
integer; var gcd:integer);
  {“простой алгоритм”
      НОД(a,b)
  Вход: числа a,b;
  Выход: НОД(a,b) – gcd}

```

```

begin {a>0 and b>0}
  while a<>b do
    if a>b then a:=a-b
    else b:=b-a;
  gcd :=a
end;

```

```

procedure gcd_Evkidl(a,b:
integer; var gcd:integer);
  {алгоритм Евклида
      НОД (a,b)
  Вход: числа a,b;
  Выход: НОД(a,b) – gcd }

```

```

begin
  repeat
    r:=a mod b;
    a:=b; b:=r
  until b=0;
  gcd:=a
end;

```

Когда необходимо вычислить **НОД** нескольких чисел можно применить несколько методов:

1) распространение алгоритма Евклида, базирующегося на следующих свойствах:

a)  $\text{НОД}(0, \dots, 0, a, 0, \dots, 0) = a;$

b)  $\text{НОД}(a_1, \dots, a_i, \dots, a_n) =$

$\text{НОД}(a_1 \bmod a_i, \dots, a_i, \dots, a_n \bmod a_i)$  при  $a_i \neq 0.$

2) метод заключается в повторном применении алгоритма Евклида для двух целых чисел.

Он основан на следующем свойстве:

$$\text{НОД}(a_1, \dots, a_n) = \text{НОД}(a_1, \text{НОД}(a_2, \dots, a_n)),$$

которое порождает рекурсивный алгоритм вычисления НОД. Именно

$$\text{НОД}(a_1, \dots, a_n) = \text{НОД}(\text{НОД}(a_1, a_2), a_3, \dots, a_n),$$

что является основой соответствующего итеративного алгоритма.

**Теорема Дирихле.** Если  $a$  и  $b$  два натуральных числа, выбранные случайно, то вероятность того, что они взаимно простые равна

$$\frac{6}{\pi^2} \approx 0,607927$$

**Теорема Ламе.** Число итераций, необходимых для вычисления НОД( $a, b$ ),  $a > b > 0$ , мажорируется 5-кратным числом десятичных знаков наименьшего из этих двух чисел. Более формально, если  $n$  является искомым числом итераций, то

$$n \leq 5([\log_{10} b] + 1) \quad \text{или} \quad n \leq 5[\log_{10}(b + 1)].$$

**Главный результат** – сложность алгоритма Евклида для целых чисел логарифмическая по отношению к наименьшему из двух чисел. В оценке Ламе коэффициент 5 оптимален, но мажорирующая функция ( $O(\log b)$ ) таковой не является.

# Расширенный алгоритм Евклида

Алгоритм, примененный к паре чисел  $a, b$  порождает последовательность  $(r_i)_{0 \leq i \leq n+1}$  такую, что

$$r_{i-1} = r_i q_i + r_{i+1} \text{ для } 1 \leq i \leq n, \text{ где } r_0 = a, r_1 = b, r_{n+1} = 0.$$

Из этих формул легко получается рекуррентная последовательность:

$$\begin{cases} u_0 = 1, & v_0 = 0, & r_0 = a, \\ u_1 = 0, & v_1 = 1, & r_1 = b, \\ u_{i+1} = u_{i-1} - q_i u_i, & v_{i+1} = v_{i-1} - q_i v_i, & r_{i+1} = r_{i-1} - q_i r_i, \end{cases}$$

из которой теперь следует классический результат

$$r_n = \text{НОД}(a, b) = u_n a + v_n b.$$