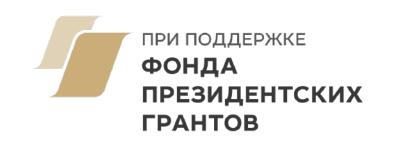
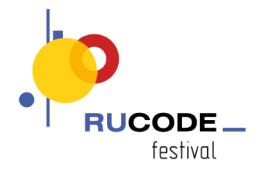
## Алгоритм Евклида. Расширенный алгоритм Евклида

**Урок 2.3** 











## В этом видео\_

- Применение свойств остатков
- НОД
- Алгоритм Евклида
- Расширенный алгоритм Евклида









## Наибольший общий делитель: определения\_

- Наибольший общий делитель а и b обозначается как (a,b)
- (a,b) = d обозначает, что d > 0,  $a = a_1 d$ ,  $b = b_1 d$ , при этом d максимально

#### Свойства наибольшего общего делителя:

- (a,a) = a. (a,0) = a. (a,1) = 1. (-a,b) = (a,b)
- Если p простое, то (p,a) = p, если  $a \mod p = 0$ , иначе 1
- Если (a,b) = 1, то a и b называются взаимно простыми









## Наибольший общий делитель и остатки\_

- Пусть d = (b,a mod b), то есть b = xd, a mod b = yd.
   a = bq + (a mod b) = xdq + yd = d(xq + y), то есть d делитель a
- Пусть d не наибольший, то есть существует
   D > d: a = Dx<sub>1</sub>, b = Dy<sub>1</sub>

Тогда a mod  $b = a - bq = Dx_1 - Dy_1q = D(x_1 - y_1q),$  тем самым получили, что и b, и a mod b делятся на D — противоречит выбору d

To есть  $(a,b) = (b,a \mod b)$ 









## Алгоритм Евклида: описание\_

#### Пусть a ≥ b (если нет, переставим)

- **1.** Если b = 0, то (a,0) = a, завершаем работу и выдаём а как ответ
- **2.** При b > 0 переходим от (a,b) к (b, a mod b) и далее на пункт **1**









## Алгоритм Евклида: описание\_

#### Пусть a ≥ b (если нет, переставим)

- **1.** Если b = 0, то (a,0) = a, завершаем работу и выдаём а как ответ
- **2.** При b > 0 переходим от (a,b) к (b, a mod b) и далее на пункт **1**

Так как для положительных A и B 0 ≤ A mod B < B, то после любого шага, начиная с первого, число заменяется строго меньшим и неотрицательным.

Значит, мы за конечное число шагов завершим работу и получим ответ









### Алгоритм Евклида: код\_









```
1 int gcd (int a, int b) // предполагаем, что а неотрицательно, b положительно
2 {
3  if (a < b) return gcd (b, a);
4  if (b == 0) return a;
5  return gcd (b, a % b);
6 }
7 // вместо int может стоять любой целочисленный тип</pre>
```

## Алгоритм Евклида: оценка скорости снизу\_

- Лемма: докажем, что если A ≥ B, то A mod B < A/2</li>
- $A = zB + A \mod B$ . Так как  $A \ge B$ , то  $z \ge 1$  (иначе  $A = zA + A \mod B \le A \mod B < B$ )
- Тогда A mod B < B ≤ zB, значит, A = zB + A mod B > 2(A mod B)
   и A mod B < A/2</li>

#### Сделаем два шага алгоритма Евклида:

- (a,b) переходит в (a mod b, b mod (a mod b)), то есть за два шага оба элемента уменьшились более, чем вдвое
- Сложность не выше **O(log(a))**









## Алгоритм Евклида: оценка скорости снизу\_

- Пусть  $F_i$  i-е число Фибоначчи. Посчитаем ( $F_n$ ,  $F_{n-1}$ )
  Так как  $F_n = F_{n-1} + F_{n-2}$  и  $F_{n-1} > F_{n-2}$  при n > 2, то  $F_{n-2} = F_n \mod F_{n-1}$  ( $F_n$ ,  $F_{n-1}$ ) = ( $F_{n-1}$ ,  $F_{n-2}$ ) = ( $F_{n-2}$ ,  $F_{n-3}$ ) ...
- Но  $F_x = F_{x-1} + F_{x-2} = (F_{x-2} + F_{x-3}) + F_{x-2} = 2F_{x-2} + F_{x-3} \le 3F_{x-2}$ , так что за два шага оба элемента уменьшились не более, чем втрое не быстрее O(log(a))

Итоговая сложность: O(log(a))









## Свойства наибольшего общего делителя\_

- Для любого шага алгоритма Евклида  $(r_1, r_2) = (r_2, r_1 \mod r_2)$  справедливо  $(kr_1, kr_2) = (kr_2, kr_1 \mod kr_2) = (kr_2, k (r_1 \mod r_2))$ , поэтому (ka, kb) = k(a, b)
- Докажем лемму: если ab делится на простое p, то хотя бы одно из a или b делится на p
- Пусть b не делится на p, то есть (b,p) = 1. Рассмотрим (ap,ab). ар и ab делятся на p, то есть (ap,ab) делится на p
   Но (ap,ab) = a(p,b) = a \* 1 = a, то есть а делится на p









# Какие значения может принимать ах + by?\_

- Так как  $ax+by = a_1(a,b)x + b_1(a,b)y$ , то ax + by всегда делится на (a,b). Значит, все значения ax + by имеют вид k(a,b)
- Пусть  $r_1 = ax_1 + by_1$ ,  $r_2 = ax_2 + by_2$ ,  $r_1 > r_2$ .

  Тогда  $r_1 \mod r_2 = r_1 qr_2 = ax_1 + by_1 q(ax_2 + by_2) = a(x_1 qx_2) + b(y_1 qy_2) вид ах + by$
- Следствие: так как (a,b) = (a \* 1 + 0 \* b, a \* 0 + 1 \*b),
   на всех шагах алгоритма Евклида числа имеют вид ах + by,
   то есть (a,b) = aX + bY и k(a,b) = a(kX) + b(kY)

Значит, ax+by принимает все значения вида k(a,b), и только такие значения









### Описание\_

- Пусть  $\mathbf{x_1}$ ,  $\mathbf{y_1}$ ,  $\mathbf{x_2}$ ,  $\mathbf{y_2}$  коэффициенты при а и b для большего  $(\mathbf{r_1})$  и меньшего  $(\mathbf{r_2})$  чисел на текущем шаге алгоритма Евклида
- В начале: (a,b) = (1 \* a + 0 \* b, 0 \* a + 1 \* b)и  $x_1 = 1$ ,  $y_1 = 0$ ,  $x_2 = 0$ ,  $y_2 = 1$
- Шаг: (r<sub>1</sub>, r<sub>2</sub>) переходит в (r<sub>2</sub>, r<sub>1</sub> mod r<sub>2</sub>)
  - Считаем  $x = x_1 q x_2$ ,  $y = y_1 q y_2$ , где  $q = [r_1 / r_2]$
  - Переприсваиваем  $x_1 = x_2$ ,  $y_1 = y_2$ ,  $x_2 = x$ ,  $y_2 = y$
- Переходим к следующему шагу алгоритма Евклида
  - В итоге при остановке в  $x_1$  и  $y_1$  будут требуемые X и Y









## Расширенный алгоритм Евклида\_

- Пусть  $\mathbf{x}_1$ ,  $\mathbf{y}_1$ ,  $\mathbf{x}_2$ ,  $\mathbf{y}_2$  коэффициенты при а и b для большего  $(\mathbf{r}_1)$  и меньшего  $(\mathbf{r}_2)$  чисел на текущем шаге алгоритма Евклида
- В начале: (a,b) = (1 \* a + 0 \* b, 0 \* a + 1 \* b) и  $x_1 = 1$ ,  $y_1 = 0$ ,  $x_2 = 0$ ,  $y_2 = 1$
- Шаг: (r<sub>1</sub>, r<sub>2</sub>) переходит в (r<sub>2</sub>, r<sub>1</sub> mod r<sub>2</sub>)
  - Считаем  $x = x_1 q x_2$ ,  $y = y_1 q y_2$ , где  $q = [r_1 / r_2]$
  - Переприсваиваем  $x_1 = x_2$ ,  $y_1 = y_2$ ,  $x_2 = x$ ,  $y_2 = y$
- Переходим к следующему шагу алгоритма Евклида

В итоге при остановке в  $x_1$  и  $y_1$  будут требуемые X и Y









## Расширенный алгоритм Евклида: код

```
blu
```









```
1 int extgcd (int r1, int r2, int &x1, int &y1, int &x2, int &y2, bool first=true) {
2  if (r1<r2)  return extgcd (r2, r1, y1, x1, y2, x2, first); // переставляем при a<b
3  if (first) { // если первый вызов
4  x1=1;y1=0;x2=0;y2=1; } // инициализируем x_i и y_i для (a,b)
5  if (r2==0) return r1;
6  int x = x1 - (r1 / r2) * x2; // вычисляем коэффициенты для r1 % r2
7  int y = y1 - (r1 / r2) * y2;
8  x1 = x2; x2 = x; y1 = y2; y2 = y; // меняем значения переменных
9  return extgcd (r2, r1 % r2, x1, y1, x2, y2, false);
10 }
```

## Расширенный алгоритм Евклида: применение\_

- Вызов: extgcd (a,b,x,y,tX,tY), где tX и tY временные переменные
- Есть уравнение ax + by = c линейное диофантово уравнение

#### Найдём одно решение:

Если с не делится на (a,b) — решений нет (слева делится, справа нет)

Иначе расширенный алгоритм Евклида даст

$$x_0$$
 и  $y_0$ :  $ax_0 + by_0 = (a,b)$ 

A затем  $x = x_0 c / (a,b), y = y_0 c / (a,b)$ 









## Подведем итог\_

- Научились находить НОД
- Расширенный алгоритм Евклида







