Алгоритм Евклида. Расширенный алгоритм Евклида

Урок 2.3

МФТИ
Московский физико-технический институт

При поддержке фонда президентских грантов

RUCODE festival

Фонд развития Физтех-школ

В этом видео_

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

Наибольший общий делитель: определения_

- *Наибольший общий делитель* a 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 **$a \bmod p = 0$**, иначе 1
- Если $(a,b) = 1$, то a и b называются *взаимно простыми*

Наибольший общий делитель и остатки_

- Пусть $d = (b, a \bmod b)$, то есть $b = xd$, $a \bmod b = yd$.
  $a = bq + (a \bmod b) = xdq + yd = d(xq + y)$, то есть d — делитель a

- Пусть d — не наибольший, то есть существует
  $D > d$: $a = Dx_1$, $b = Dy_1$

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

  То есть $\mathbf{(a,b) = (b, a \bmod b)}$

Алгоритм Евклида: описание_

Пусть $a \geq b$ (если нет, переставим)

1. Если $b = 0$, то $(a,0) = a$, завершаем работу
и выдаём a как ответ

2. При $b > 0$ переходим от (a,b) к $(b, a \bmod b)$
и далее на пункт 1

Алгоритм Евклида: описание_

Пусть $a \geq b$ (если нет, переставим)

1. Если $b = 0$, то $(a,0) = a$, завершаем работу
и выдаём a как ответ

2. При $b > 0$ переходим от (a,b) к $(b, a \bmod b)$
и далее на пункт 1

Так как для положительных A и B $0 \leq A \bmod B < B$, 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_

```
int gcd (int a, int b)  // предполагаем, что a неотрицательно, b положительно
{
 if (a < b) return gcd (b, a);
 if (b == 0) return a;
 return gcd (b, a % b);
}
// вместо int может стоять любой целочисленный тип
```

Алгоритм Евклида: оценка скорости снизу_

- Лемма: докажем, что если $A \geq B$, то $A \bmod B < A/2$
- $A = zB + A \bmod B$. Так как $A \geq B$, то $z \geq 1$ (иначе $A = zA + A \bmod B \leq A \bmod B < B$)
- Тогда $A \bmod B < B \leq zB$, значит, $A = zB + A \bmod B > 2(A \bmod B)$ и $A \bmod B < A/2$

Сделаем два шага алгоритма Евклида:

- (a,b) переходит в $(a \bmod b, b \bmod (a \bmod b))$, 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 $F_{n-2} = F_n \bmod F_{n-1}$
  $(F_n,F_{n-1}) = (F_{n-1},F_{n-2}) = (F_{n-2},F_{n-3}) \ldots$

- 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 $(r_1,r_2) = (r_2,r_1 \bmod r_2)$ справедливо $(kr_1,kr_2) = (kr_2,kr_1 \bmod kr_2) = (kr_2,k(r_1 \bmod r_2))$, 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). ap и ab делятся на p, то есть (ap,ab) делится на p
Но $(ap,ab) = a(p,b) = a * 1 = a$, то есть a делится на p

Какие значения может принимать ax + 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 $r_1 \bmod r_2 = r_1 – qr_2 = ax_1 + by_1 – q(ax_2 + by_2) =$
  $a (x_1 - qx_2) + b (y_1 - q y_2)$ — вид $ax + by$

- Следствие: так как $(a,b) = (a * 1 + 0 * b, a * 0 + 1 *b)$, на всех шагах алгоритма Евклида числа имеют вид $ax + by$, 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 x_1, y_1, x_2, y_2 — коэффициенты при a и b для большего (r_1) и меньшего (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_1, r_2) переходит в $(r_2, r_1 \bmod r_2)$
  - 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 x_1, y_1, x_2, y_2 — коэффициенты при a и b для большего (r_1) и меньшего (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_1, r_2) переходит в $(r_2, r_1 \bmod r_2)$
  - 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_

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

МФТИ
ПРИ ПОДДЕРЖКЕ ФОНДА ПРЕЗИДЕНТСКИХ ГРАНТОВ
RUCODE festival
Фонд развития Физтех-школ

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

Иначе расширенный алгоритм Евклида даст x_0 и y_0: $ax_0 + by_0 = (a,b)$

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

Подведем итог_

- Научились находить НОД
- Расширенный алгоритм Евклида