

Лабораторная работа 4

Вычисление наибольшего общего делителя

Пологов Владислав Александрович

2022 Москва

RUDN University, Moscow, Russian Federation

Цель работы

Реализовать алгоритмы вычисления наибольшего общего делителя:

1. Алгоритм Евклида
2. Бинарный Алгоритм Евклида
3. Расширенный алгоритм Евклида
4. Расширенный бинарный алгоритм Евклида

Описание реализации

Для реализации алгоритмов использовались средства языка Python.

Реализация

На вход мы подаём два целых числа a и b . На выходе получаем d - НОД. Алгоритм Евклида и его реализация на Python приведены на рисунке 1. (рис. -fig. 1)

Алгоритм Евклида

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, i \leftarrow 1$.
2. Найти остаток r_{i+1} от деления r_{i-1} на r_i .
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i$. В противном случае положить $i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d .

```
def evklid(a, b, d):  
    for i in range(1, b + 1):  
        if a % i == 0 and b % i == 0:  
            d = i  
        else:  
            continue  
    return d
```

Figure 1: Алгоритм Евклида

Для реализации бинарного алгоритма Евклида использовалась дополнительная переменная g . Данный алгоритм и его реализация на Python представлены на рисунке 2. (рис. -fig. 2)

Бинарный алгоритм Евклида

1. Положить $g \leftarrow 1$.
2. Пока оба числа a и b четные, выполнять $a \leftarrow \frac{a}{2}$, $b \leftarrow \frac{b}{2}$, $g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b .
3. Положить $u \leftarrow a$, $v \leftarrow b$.
4. Пока $u \neq 0$ выполнять следующие действия:
 - 4.1. Пока u четное, полагать $u \leftarrow \frac{u}{2}$.
 - 4.2. Пока v четное, полагать $v \leftarrow \frac{v}{2}$.
 - 4.3. При $u \geq v$ положить $u \leftarrow u - v$. В противном случае положить $v \leftarrow v - u$.
5. Положить $d \leftarrow gv$.
6. Результат: d

```
def binary_evklid(a, b):  
    g = 1  
    while (a % 2 == 0 and b % 2 == 0):  
        a, b, g = a // 2, b // 2, g * 2  
    while a != 0:  
        while a % 2 == 0:  
            a = a // 2  
        while b % 2 == 0:  
            b = b // 2  
        if a >= b:  
            a = a - b  
        else:  
            b = b - a  
    return b * g
```

В расширенном алгоритме Евклида также необходимо соблюдение следующего условия:

$$a * x + b * y = d$$

Расширенный алгоритм Евклида и его реализация на Python представлены на рисунке 3. (рис. -fig. 3)

Расширенный алгоритм Евклида

1. Положить $r_0 \leftarrow a, r_1 \leftarrow b, x_0 \leftarrow 1, x_1 \leftarrow 0, y_0 \leftarrow 0, y_1 \leftarrow 1, i \leftarrow 1$.
2. Разделить с остатком r_{i-1} на r_i : $r_{i-1} = q_i r_i + r_{i+1}$.
3. Если $r_{i+1} = 0$, то положить $d \leftarrow r_i, x \leftarrow x_i, y \leftarrow y_i$. В противном случае положить $x_{i+1} \leftarrow x_{i-1} - q_i x_i, y_{i+1} \leftarrow y_{i-1} - q_i y_i, i \leftarrow i + 1$ и вернуться на шаг 2.
4. Результат: d, x, y .

```
def extend_evklid(a, b):  
    r0 = a  
    r1 = b  
    x0 = 1  
    x1 = 0  
    y0 = 0  
    y1 = 1  
    i = 1  
  
    while r1 != 0:  
        q = r0 // r1  
        r0 = r0 % r1  
        r0, r1 = r1, r0  
  
        x0 -= q * x1  
        x0, x1 = x1, x0  
  
        y0 -= q * y1  
        y0, y1 = y1, y0  
    return r0, x0, y0
```

Расширенный бинарный алгоритм Евклида

В расширенном бинарном алгоритме Евклида сочетаются методы используемые в расширенном и бинарном алгоритмах.

Расширенный бинарный алгоритм Евклида представлен на рисунке 4. (рис. -fig. 4)

Код расширенного бинарного алгоритма Евклида представлен на рисунке 5. (рис. -fig. 5)

Расширенный бинарный алгоритм Евклида

1. Положить $g \leftarrow 1$.
2. Пока числа a и b четные, выполнять $a \leftarrow \frac{a}{2}, b \leftarrow \frac{b}{2}, g \leftarrow 2g$ до получения хотя бы одного нечетного значения a или b .
3. Положить $u \leftarrow a, v \leftarrow b, A \leftarrow 1, B \leftarrow 0, C \leftarrow 0, D \leftarrow 1$.
4. Пока $u \neq 0$ выполнять следующие действия:
 - 4.1. Пока u четное:
 - 4.1.1. Положить $u \leftarrow \frac{u}{2}$.
 - 4.1.2. Если оба числа A и B четные, то положить $A \leftarrow \frac{A}{2}, B \leftarrow \frac{B}{2}$. В противном случае положить $A \leftarrow \frac{A+b}{2}, B \leftarrow \frac{B-a}{2}$.
 - 4.2. Пока v четное:
 - 4.2.1. Положить $v \leftarrow \frac{v}{2}$.
 - 4.2.2. Если оба числа C и D четные, то положить $C \leftarrow \frac{C}{2}, D \leftarrow \frac{D}{2}$. В противном случае положить $C \leftarrow \frac{C+b}{2}, D \leftarrow \frac{D-a}{2}$.
 - 4.3. При $u \geq v$ положить $u \leftarrow u - v, A \leftarrow A - C, B \leftarrow B - D$. В противном случае положить $v \leftarrow v - u, C \leftarrow C - A, D \leftarrow D - B$.
5. Положить $d \leftarrow gv, x \leftarrow C, y \leftarrow D$.
6. Результат: d, x, y .

Расширенный бинарный алгоритм Евклида

```
def extend_binary_evklid(a, b):  
    g = 1  
    while (a % 2 == 0 and b % 2 == 0):  
        a, b, g = a // 2, b // 2, g * 2  
  
    u, v = a, b  
    A, B, C, D = 1, 0, 0, 1  
  
    while u != 0:  
        while u % 2 == 0:  
            u //= 2  
            if A % 2 == 0 and B % 2 == 0:  
                A, B = A//2, B//2  
            else:  
                A = (A + b) // 2  
                B = (B - a) // 2  
        while v % 2 == 0:  
            v //= 2  
            if C % 2 == 0 and D % 2 == 0:  
                C, D = C//2, D//2  
            else:  
                C = (C + b) // 2  
                D = (D - a) // 2  
        if u >= v:  
            u -= v  
            A -= C  
            B -= D  
        else:  
            v -= u  
            C -= A  
            D -= B  
  
    d = g*v  
    x = C  
    y = D  
    return d, x, y
```

Вывод

- Реализовали следующие алгоритмы для нахождения НОД:
 1. Алгоритм Евклида
 2. Бинарный Алгоритм Евклида
 3. Расширенный алгоритм Евклида
 4. Расширенный бинарный алгоритм Евклида.

