

Отчёт по лабораторной работе №4

Вычисление наибольшего общего делителя

Волкова Дарья Александровна НПМмд-02-21

Содержание

Цель работы	1
Теоретические сведения	1
Алгоритм Евклида	2
Бинарный алгоритм Евклида	2
Расширенный алгоритм Евклида	3
Расширенный бинарный алгоритм Евклида	3
Выполнение работы	4
Реализация алгоритмов на языке Python	4
Контрольный пример	6
Выводы	9
Список литературы	10

Цель работы

Изучение понятия наибольшего общего делителя и его вычисление. Изучение алгоритмов вычисления наибольшего общего делителя, таких как алгоритм Евклида, бинарный алгоритм Евклида, расширенный алгоритм Евклида и расширенный бинарный алгоритм Евклида, а также их программная реализация.

Теоретические сведения

Для начала разберемся, что такое общий делитель. У целого числа может быть несколько делителей. Делитель натурального числа — это такое целое натуральное число, на которое делится данное число без остатка. Если у натурального числа больше двух делителей, его называют составным.

Общий делитель нескольких целых чисел — это такое число, которое может быть делителем каждого числа из указанного множества. Например, у чисел 12 и 8 общими делителями будут 4 и 1.

Любое число можно разделить на 1 и на само себя. Значит, у любого набора целых чисел будет как минимум два общих делителя.

Наибольшим общим делителем двух чисел a и b называется наибольшее число, на которое a и b делятся без остатка. Для записи может использоваться аббревиатура НОД. Для двух чисел можно записать вот так: НОД (a , b). Например, для 4 и 16 НОД будет 4. Наибольший общий делитель двух и более натуральных чисел – это наибольшее из натуральных чисел, на которое делится каждое из данных чисел.

Алгоритм Евклида

Алгоритм Евклида – это алгоритм нахождения наибольшего общего делителя пары целых чисел.

Алгоритм Евклида — один из наиболее ранних численных алгоритмов в истории. Название было дано в честь греческого математика Евклида, который впервые дал ему описание в книгах «Начала». Изначально назывался «взаимным вычитанием», так как его принцип заключался в последовательном вычитании из большего числа меньшего, пока в результате не получится ноль. Сегодня чаще используется взятие остатка от деления вместо вычитания, но суть метода сохранилась.

Простейшим случаем применения данного алгоритма является поиск наибольшего общего делителя для пары положительных целых чисел. Евклид, автор этого метода, предполагал его использование только для натуральных чисел и геометрических величин. Но позже алгоритм был обобщен и для других групп математических объектов, что привело к появлению такого понятия, как евклидово кольцо.

Алгоритм Евклида — это алгоритм, основная функция которого заключается в поиске наибольшего общего делителя для двух целых чисел.

Последовательность нахождения НОД при помощи деления:

1. Большее число делим на меньшее.
2. Если делится без остатка, то меньшее число и есть НОД (следует выйти из цикла).
3. Если есть остаток, то большее число заменяем на остаток от деления.
4. Переходим к пункту 1.

Последовательность нахождения НОД при помощи вычитания:

1. Из большего числа вычитается меньшее.
2. Если результат вычитания:
 - 2.1. равен 0, то числа равны друг другу и являются НОД;
 - 2.2. не равен 0, в таком случае большее число заменяется на результат вычитания.
3. Переход к пункту 1.

Бинарный алгоритм Евклида

Бинарный алгоритм вычисления НОД, как понятно из названия, находит наибольший общий делитель, а именно НОД двух целых чисел. В эффективности данный алгоритм превосходит метод Евклида, что связано с использованием сдвигов, то есть операций деления на степень 2-ки, в нашем случае на 2. Но в тоже время бинарный алгоритм уступает алгоритму Евклида в простоте реализации.

Интересен тот факт, что алгоритм был известен еще в Китае 1-го века н. э., но годом его обнародования оказался лишь 1967, когда израильский программист и физик Джозеф Стейн опубликовал алгоритм. Ввиду этого встречается альтернативное название метода – алгоритм Стейна.

Теперь рассмотрим этапы работы алгоритма:

1. Инициализируем переменную k значением 1. Ее задача – подсчитывать «несоразмерность», полученную в результате деления. В то время как A и B сокращаются вдвое, она будет увеличиваться вдвое;
2. Пока A и B одновременно не равны нулю, выполняем
 - 2.1. если A и B – четные числа, то делим надвое каждое из них: $A \leftarrow A/2$, $B \leftarrow B/2$, а k увеличивать вдвое: $k \leftarrow k*2$, до тех пор, пока хотя бы одно из чисел A или B не станет нечетным;
 - 2.2. если A – четное, а B – нечетное, то делим A , пока возможно деление без остатка;
 - 2.3. если B – четное, а A – нечетное, то делим B , пока возможно деление без остатка;
 - 2.4. если $A \geq B$, то заменим A разностью A и B , иначе B заменим разностью B и A .
3. После выхода из 2-ого пункта, остается вернуть в качестве результата произведение B и k : $\text{НОД}(A, B) = B*k$.

Расширенный алгоритм Евклида

Если стандартный алгоритм Евклида определяет наибольший общий делитель числовой пары a и b , то расширенный алгоритм находит не только наименьший общий делитель, но и коэффициенты x и y , для которых выполняется равенство:

$$a*x + b*y = \text{НОД}(a, b)$$

То есть с его помощью определяются коэффициенты, при помощи которых наименьший общий делитель пары чисел можно выразить через саму пару этих чисел. Для определения этих коэффициентов необходимо получить формулы их изменения при переходе от пары (a, b) к паре $(b \% a, a)$. Здесь символ процентов обозначает остаток от деления. Предположим, что существует решение (X, Y) для вновь созданной пары $(b \% a, a)$:

$$b \% a * X + a * Y = g$$

и надо найти решение (x, y) для искомой пары (a, b) :

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

Выполнив ряд преобразований, находим конечные формулы:

$$x = Y - b/a * X$$

$$y = X$$

Расширенный бинарный алгоритм Евклида

Вход. Целые числа a, b - $0 < b \leq a$.

Выход. $d = \text{НОД}(a, b)$ - такие целые числа x, y , что $ax + by = d$.

Выполнение работы

Реализация алгоритмов на языке Python

1. Алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))
a=int(input('Введите число a больше или равно b: '))

r_i=[a,b]
i=1

while True:
    r=r_i[i-1]%r_i[i]
    if r==0:
        d=r_i[i]
        break
    else:
        i+=1
        r_i.append(r)

print('a=',a,', b=',b,', НОД(a,b)=',d)
```

2. Бинарный алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))
a=int(input('Введите число a больше или равно b: '))

A=a
B=b
g=1

while (a % 2 == 0) and (b % 2 == 0):
    a=a/2
    b=b/2
    g=2*g
u=a
v=b
while u!=0:
    while u%2 == 0:
        u=u/2
    while v%2 == 0:
        v=v/2
    if u>v:
        u=u-v
    else:
        v=v-u
d=g*v

print('a=',A,', b=',B,', НОД(a,b)=',d)
```

3. Расширенный алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))
a=int(input('Введите число a больше или равно b: '))
```

```
r=[a,b]
x_i=[1,0]
y_i=[0,1]
i=1
```

```
while True:
    if r[i-1] % r[i] == 0:
        d = r[i]
        x = x_i[i]
        y = y_i[i]
        break
    else:
        q = r[i-1] // r[i]
        x_i.append(x_i[i-1]-q*x_i[i])
        y_i.append(y_i[i-1]-q*y_i[i])
        r.append(r[i-1] % r[i])
        i=i+1
```

```
print('a=',a,', b=',b,', НОД(a,b)=',d)
```

4. Расширенный бинарный алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))
a=int(input('Введите число a больше или равно b: '))
```

```
g=1
```

```
while a%2 == 0 and b%2 == 0:
    a=a/2
    b=b/2
    g=2*g
```

```
u=a
v=b
A=1
B=0
C=0
D=1
```

```
while u!=0:
    while u%2 == 0:
        u=u/2
        if A%2 == 0 and B%2 == 0:
            A=A/2
            B=B/2
        else:
            A=(A+b)/2
```

$B=(B-a)/2$

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

print('НОД(a,b)=',d)
```

Контрольный пример

1. Алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))
a=int(input('Введите число a больше или равно b: '))
```

Введите число b больше 0: 45
Введите число a больше или равно b: 60

```
r_i=[a,b]
i=1
```

```
while True:
    r=r_i[i-1]%r_i[i]
    if r==0:
        d=r_i[i]
        break
    else:
        i+=1
    r_i.append(r)
```

```
print('a=',a,', b=',b,', НОД(a,b)=',d)
```

a= 60 , b= 45 , НОД(a,b)= 15

Пример работы алгоритма Евклида

```
: # 2. Бинарный алгоритм Евклида
```

```
: b=int(input('Введите число b больше 0: '))  
a=int(input('Введите число a больше или равно b: '))
```

Введите число b больше 0: 45
Введите число a больше или равно b: 60

```
: A=a  
B=b  
g=1
```

```
: while (a % 2 == 0) and (b % 2 == 0):  
    a=a/2  
    b=b/2  
    g=2*g  
u=a  
v=b  
while u!=0:  
    while u%2 == 0:  
        u=u/2  
    while v%2 == 0:  
        v=v/2  
    if u>=v:  
        u=u-v  
    else:  
        v=v-u  
d=g*v
```

```
: print('a=',A,', b=',B,', НОД(a,b)=' ,d)
```

a= 60 , b= 45 , НОД(a,b)= 15.0

Пример работы бинарного алгоритма Евклида

3. Расширенный алгоритм Евклида

```
b=int(input('Введите число b больше 0: '))  
a=int(input('Введите число a больше или равно b: '))
```

Введите число b больше 0: 45
Введите число a больше или равно b: 60

```
r=[a,b]  
x_i=[1,0]  
y_i=[0,1]  
i=1
```

```
while True:  
    if r[i-1] % r[i] == 0:  
        d = r[i]  
        x = x_i[i]  
        y = y_i[i]  
        break  
    else:  
        q = r[i-1] // r[i]  
        x_i.append(x_i[i-1]-q*x_i[i])  
        y_i.append(y_i[i-1]-q*y_i[i])  
        r.append(r[i-1] % r[i])  
        i=i+1
```

```
print('a=',a,', b=',b,', НОД(a,b)=',d)
```

a= 60 , b= 45 , НОД(a,b)= 15

Пример работы расширенного алгоритма Евклида


```
: # 4. Расширенный бинарный алгоритм Евклида
```

```
: b=int(input('Введите число b больше 0: '))  
a=int(input('Введите число a больше или равно b: '))
```

```
Введите число b больше 0: 45  
Введите число a больше или равно b: 60
```

```
: g=1
```

```
: while a%2 == 0 and b%2 == 0:  
    a=a/2  
    b=b/2  
    g=2*g
```

```
: u=a  
v=b  
A=1  
B=0  
C=0  
D=1
```

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

```
: print('НОД(a,b)=',d)
```

```
НОД(a,b)= 15.0
```

Пример работы расширенного бинарного алгоритма Евклида

Выводы

В ходе выполнения работы удалось изучить понятие наибольшего общего делителя и его вычисление. Также изучить алгоритмы вычисления наибольшего общего делителя, такие как алгоритм Евклида, бинарный алгоритм Евклида, расширенный алгоритм Евклида и расширенный бинарный алгоритм Евклида, а также реализовать данные алгоритмы программно на языке Python.

Список литературы

1. Бинарный алгоритм вычисления НОД
2. Как найти НОД двух чисел по алгоритму Евклида
3. Наибольший общий делитель. Алгоритм Евклида
4. Расширенный алгоритм Евклида
5. Расширенный алгоритм Евклида