Отчёт по лабораторной работе №4

Дисциплина: Математические основы защиты информации и информационной безопасности

Полиенко Анастасия Николаевна, НПМмд-02-23

Содержание

Список иллюстраций

1 Цель работы

Изучить алгоритмы вычисления наибольшего общего делителя.

2 Задание

Реализовать алгоритм Евклида в четырёх его вариациях:

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

3 Теоретическое введение

Пусть числа a и b целые и $b \neq 0$. Разделить a на b с остатком - значит представить a в виде $a = qb + r$, где $q, r \in \mathbb{Z}$ и $0 \leq r \leq |b|$. Число q называется неполным частным, число r - неполным остатком от деления a на b.

Целое число $d \neq 0$ называется *наибольшим общим делителем* целых чисел $a_1, a_2, \dots, a_k$ (обозначается $d =$ НОД($a_1, a_2, \dots, a_k$)), если выполняются следующие условия:

1. Каждое из чисел $a_1, a_2, \dots, a_k$ делится на d;
2. Если $d_1 \neq 0$ - другой общий делитель чисел $a_1, a_2, \dots, a_k$, то d_1 делится на d.

Например, НОД(12345, 24690) = 12345, НОД(12345, 54321) = 3, НОД(12345, 12541) = 1.

Ненулевые целые числа a и b называются *ассоциированными* (обозначается a b), если a делится на b и b делится на a.

Для любых целых чисел $a_1, a_2, \dots, a_k$ существует наибольший общий делитель d и его можно представить в виде *линейной комбинации* этих чисел:

$d = c_1 a_1 + c_2 a_2 + \dots + c_k a_k, c_i \in \mathbb{Z}$

Например, НОД чисел 91, 105, 154 равен 7. В качестве линейного представления можно взять

$7 = 7 \cdot 91 - 6 \cdot 105 + 0 \cdot 154,$

либо

$7 = 4 \cdot 91 + 1 \cdot 105 - 3 \cdot 154$

Целые числа $a_1, a_2, \ldots, a_k$ называются *взаимно простыми в совокупности*, если НОД($a_1, a_2, \ldots, a_k$) = 1. Целые числа a и b называются *взаимно* простыми, если НОД(*a, b*) = 1.

Целые числа $a_1, a_2, \ldots, a_k$ называются *попарно взаимно простыми*, если НОД(a_i, a_j) = 1 для всех $1 \leq i \neq j \leq k$.

Более подробно см. в [1–6].

4 Выполнение лабораторной работы

1. Реализуем алгоритм Евклида нахождения наименьшего общего делителя (рис. 4.1).

```
a = int(input())
b = int(input())
gcd = -1

r_0 = a; r_1 = b
r_2 = -1
while r_2 != 0:
    gcd = r_2
    r_2 = r_0 % r_1
    r_0 = r_1
    r_1 = r_2
print("НОД(", a, ", ", b, ") = ", gcd, sep='')
```

```
48
36
НОД(48, 36) = 12
```

Рис. 4.1: Алгоритм Евклида

2. Реализуем бинарный алгоритм Евклида нахождения наименьшего общего делителя (рис. 4.2).

```
a_0 = int(input())
b_0 = int(input())
gcd = -1

a = a_0
b = b_0
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

gcd = g*v
print("НОД(", a_0, ", ", b_0, ") = ", gcd, sep='')
```

```
48
36
НОД(48, 36) = 12
```

Рис. 4.2: Бинарный алгоритм Евклида

3. Реализуем расширенный алгоритм Евклида нахождения наименьшего общего делителя и представления его в виде линейной комбинации чисел a и b (рис. 4.3).

```
a = int(input())
b = int(input())
gcd = -1
x = -1
y = -1

r_0 = a; r_1 = b;
x_0 = 1; x_1 = 0
y_0 = 0; y_1 = 1
r_2 = -1; q = -1; x_2 = -1; y_2 = -1

while r_2 != 0:
    print(r_2)
    gcd = r_2
    x = x_2
    y = y_2
    r_2 = r_0 % r_1
    q = r_0 // r_1
    x_2 = x_0 - q * x_1
    y_2 = y_0 - q * y_1
    r_0 = r_1
    r_1 = r_2

print("НОД(", a, ", ", b, ") = ", gcd, sep='')
print("x =", x)
print("y =", y)
print("ax + by =", a*x+b*y)
```

```
48
36
-1
12
НОД(48, 36) = 12
x = 1
y = -1
ax + by = 12
```

Рис. 4.3: Расширенный алгоритм Евклида

4. Реализуем расширенный бинарный алгоритм Евклида нахождения наименьшего общего делителя и представления его в виде линейной комбинации чисел a и b (рис. 4.4).

```python
a_0 = int(input())
b_0 = int(input())
gcd = -1
x = -1
y = -1

a = a_0
b = b_0
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

gcd = g*v
x = C
y = D
print("НОД(", a_0, ", ", b_0, ") = ", gcd, sep='')
print("x =", x)
print("y =", y)
print("ax + by =", a_0*x+b_0*y)
```

```
48
36
НОД(48, 36) = 12
x = 1
y = -1
ax + by = 12
```

Рис. 4.4: Расширенный бинарный алгоритм Евклида

5 Выводы

Изучила алгоритмы нахождения наименьшего общего делителя.

Список литературы

1. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: https://www.gnu.org/software/bash/manual/.
2. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
3. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
4. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
5. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
6. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.