

Защита лабораторной работы №4

Алгоритмы вычисления наибольшего общего делителя

Бармина Ольга

2024 September 7th

Российский университет дружбы народов, Москва, Россия

- Освоение алгоритмов вычисления наибольшего общего делителя
- Программная реализация алгоритмов вычисления НОД

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:

$$d = c_1 a_1 + c_2 a_2 + \dots + c_k a_k$$

Реализовать алгоритмы вычисления наибольшего общего делителя, такие как: - Алгоритм Евклида - Бинарный алгоритм Евклида - Расширенный алгоритм Евклида - Расширенный бинарный алгоритм Евклида

Исходные данные: - $a = 12345$ - $b = 54321$

Результат выполнения лабораторной работы

Вычисление НОД при помощи алгоритма Евклида:

```
def euclid(a,b):  
    while a!=0 and b!=0:  
        if a>=b:  
            a = a%b  
        else:  
            b = b%a  
    return a or b
```

```
a = 12345  
b = 54321  
euclid(a,b)
```

3

Figure 1: Алгоритм Евклида

Результат выполнения лабораторной работы

Вычисление НОД при помощи бинарного алгоритма Евклида:

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

bin_euclid(a,b)

3.0

Figure 2: Бинарный алгоритм Евклида

Вычисление НОД при помощи расширенного алгоритма Евклида:

```
def ext_euclid(a, b):  
    if a == 0:  
        return(b, 0, 1)  
    else:  
        d, x, y = ext_euclid(b%a,a)  
        return d, x-(b//a)*y, y
```

```
ext_euclid(a,b)
```

```
(3, -177, 1)
```

Figure 3: Расширенный алгоритм Евклида

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

```
def ext_bin_euclid(a1,b1):
    g = 1
    while a1%2 == 0 and b1%2 == 0:
        a1 /= 2
        b1 /= 2
        g *= 2
    u, v = a1, b1
    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 /= 2
                b /= 2
            else:
                a = (a + b1)//2
                b = (b - a1)//2
        while v%2 == 0:
            v /= 2
            if c%2 == 0 and d%2 == 0:
                c /= 2
                d /= 2
            else:
                c = (c + b1)//2
                d = (d - a1)//2
```

Figure 4: Расширенный бинарный алгоритм Евклида

Результат выполнения лабораторной работы

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

```
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
```

```
ext_bin_euclid(a,b)
```

```
(3.0, -14490.0, 3.0)
```

Figure 5: Расширенный бинарный алгоритм Евклида 2

Выводы

1. Изучили алгоритмы вычисления наибольшего общего делителя
2. Реализовали алгоритмы вычисления НОД