

Защита лабораторной работы №4

Алгоритмы вычисления наибольшего общего делителя

Бурдина К. П.

23 октября 2023

Российский университет дружбы народов, Москва, Россия

- * Бурдина Ксения Павловна
- * студентка группы НФИмд-02-23
- * студ. билет № 1132236896
- * Российский университет дружбы народов
- * 1132236896@rudn.ru



Вводная часть

- 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:

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

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

$$7 = 7 * 91 + (-6) * 105 + 0 * 154,$$

либо

$$7 = 4 * 91 + 1 * 105 - 3 * 154.$$

Результат выполнения лабораторной работы

Постановка задачи:

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

Исходные данные: - $a = 86415$ - $b = 12345$

Результат выполнения лабораторной работы

Вычисление НОД при помощи алгоритма Евклида:

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

```
alg_e(a, b)
```

12345

Figure 1: Алгоритм Евклида

Вычисление НОД при помощи бинарного алгоритма Евклида:

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

```
alg_e_bin(a, b)
```

```
12345
```

Figure 2: Бинарный алгоритм Евклида

Вычисление НОД при помощи расширенного алгоритма Евклида:

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

```
alg_e_ext(a, b)
```

```
(12345, 0, 1)
```

Figure 3: Расширенный алгоритм Евклида

Результат выполнения лабораторной работы

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

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

Figure 4: Расширенный бинарный алгоритм Евклида 1

Результат выполнения лабораторной работы

Вычисление НОД при помощи расширенного бинарного алгоритма Евклида:

```
if v % 2 == 0:
    v /= 2
    if (C % 2 == 0) and (D % 2 == 0):
        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)
```

Figure 5: Расширенный бинарный алгоритм Евклида 2

```
alg_e_bin_ext(a, b)
```

```
(12345, 0, 1)
```

Выводы

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