

Отчет по лабораторной работе 4

Дисциплина: Математические основы защиты информации и информационной безопасности

Дяченко З. К.

26 октября 2022

Российский университет дружбы народов, Москва, Россия

Данная лабораторная работа выполнялась мной для приобретения практических навыков нахождения наибольшего общего делителя двух целых чисел.

Цель выполнения лабораторной работы

Ознакомится и реализовать алгоритмы Евклида для нахождения НОД.

Задачи выполнения лабораторной работы

Реализовать алгоритм Евклида (рис. - fig. 1).

Алгоритм Евклида

```
In [9]: import numpy as np

In [28]: def algevc (a, b):
          if (a >= b and b > 0):
              r=[]
              r.append(a)
              r.append(b)
              i=1
              d=0
              while (d == 0):
                  r.append(r[i-1]%r[i])
                  if r[i+1]==0:
                      d=r[i]
                  else:
                      i=i+1
              return d
          else:
              print ("Ошибка, проверьте, что а больше или равно b, а b больше 0")

In [30]: algevc(100, 20)

Out[30]: 20
```

Figure 1: Реализация алгоритма Евклида

Задачи выполнения лабораторной работы

Реализовать бинарный алгоритм Евклида (рис. - fig. 2).

Бинарный алгоритм Евклида

```
In [31]: def binalgevc (a, b):  
        if (a >= b and 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):  
                if (u%2==0):  
                    u=u/2  
                if (v%2==0):  
                    v=v/2  
                if (u >= v):  
                    u=u-v  
                else:  
                    v=v-u  
            d=g*v  
            return d  
        else:  
            print ("Ошибка, проверьте, что a больше или равно b, а b больше 0")  
  
In [35]: binalgevc(100, 6)  
Out[35]: 2.0
```

Figure 2: Реализация бинарного алгоритма Евклида

Задачи выполнения лабораторной работы

Реализовать расширенный алгоритм Евклида (рис. - fig. 3).

```
Расширенный алгоритм Евклида

In [50]: def expalgevc (a, b):
        if (a >= b and b > 0):
            r=[]
            r.append(a)
            r.append(b)
            x=[]
            x.append(1)
            x.append(0)
            y=[]
            y.append(0)
            y.append(1)
            i=1
            r.append(1)
            while (r[i+1] != 0):
                r[i+1]=r[i-1]%r[i]
                q=r[i-1]//r[i]
                if (r[i+1]==0):
                    d=r[i]
                    xx=x[i]
                    yy=y[i]
                else:
                    x.append(x[i-1]-q*x[i])
                    y.append(y[i-1]-q*y[i])
                    i=i+1
            r.append(1)
            print (a, "=", xx, "+", b, "=", yy, "=", d)
            return (d, xx, yy)
        else:
            print ("Ошибка, проверьте, что а больше или равно b, а b больше 0")

In [52]: expalgevc(100, 7)

100 * -3 + 7 * 43 = 1

Out[52]: (1, -3, 43)
```

Figure 3: Реализация расширенного алгоритма Евклида

Реализовать расширенный бинарный алгоритм Евклида (рис. - fig. 4 - fig. 5).

Расширенный бинарный алгоритм Евклида

```
In [58]: def expbinalgevc (a, b):  
    if (a >= b and b > 0):  
        a_v=a  
        b_v=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
```

Figure 4: Реализация расширенного бинарного алгоритма Евклида

Задачи выполнения лабораторной работы

```
while (u != 0):
    if (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
    if (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 (a_v, "a", x, "+", b_v, "a", y, "=", d)
return(d,x,y)
else:
    print ("Ошибка, проверьте, что a больше или равно b, а b больше 0")
```

```
In [60]: expbinalgevc(100, 7)

100 * -3.0 + 7 * 43.0 = 1.0

Out[60]: (1.0, -3.0, 43.0)
```

Figure 5: Реализация расширенного бинарного алгоритма Евклида

Результатом выполнения работы стала реализация алгоритмов поиска НОД Евклида, что отражает проделанную мной работу и полученные новые знания.