# Отчет по лабораторной работе 4

Дисциплина: Математические основы защиты информации и информационной безопасности

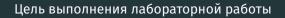
Дяченко З. К.

26 октября 2022

Российский университет дружбы народов, Москва, Россия

## Прагматика выполнения лабораторной работы

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



Ознакомится и реализовать алгоритмы Евклида для нахождения НОД.

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

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

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

Figure 2: Реализация бинарного алгоритма Евклида

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

```
Расширенный алгоритм Евклида
In [50]: def expalgevc (a. b):
             if (a >= b and b > 0):
                 r=[1
                 r annend(a)
                 r.annend(h)
                 x=[]
                 x.append(1)
                 x.append(0)
                 v=[]
                 v.append(0)
                 y.append(1)
                 3=1
                 r.append(1)
                 while (r[i+1] != 0):
                     r[i+1]=r[i-1]%r[i]
                     g=r[i-1]//r[i]
                     if (r[i+1]==0):
                         dur[i]
                         xx=x[i]
                         vv=v[i]
                         x.append(x[i-1]-q*x[i])
                         v.append(v[i-1]-q*v[i])
                         r.append(1)
                 print (a, "*", xx, "+", b, "*", yy, "=", d)
                 return (d. xx. vv)
             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
                     h=h/2
                     g=2*g
                 u=a
                 v=b
                 Δ=1
                 B=0
                 C=0
                 D=1
```

Figure 4: Реализация расширенного бинарного алгоритма Евклида

```
while (u !=0):
                     if (u%2==0):
                         u=u/2
                         if (A%2==0 and B%2==0):
                             Δ=Δ/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
                         Δ=Δ-C
                         B=B-D
                     else:
                         VEV-U
                         C=C-A
                         D=D-B
                 x=C
                 v=D
                 print (a v, "*", x, "+", b v, "*", v, "=", d)
                 return(d.x.v)
             else:
                 print ("Ошибка, проверьте, что а больше или равно b, а b больше 0")
[n [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

### Результаты выполнения лабораторной работы

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