Отчет по лабораторной работе 4

По предмету мат. основы защиты информации

Студент: Дидусь Кирилл Валерьевич, 1132223499

Группа: НПМмд-02-22

Преподаватель: Кулябов Дмитрий Сергеевич,

д-р.ф.-м.н., проф.

Москва, 2022

Содержание

# 1 Цель работы

Целью данной лабораторной работы является ознакомление с алгоритмом Евклида для нахождения НОД.

# 2 Задание

Реализовать алгоритм евклида в программном виде.

# 3 Теоритическое введение

Алгори́тм Евкли́да — эффективный алгоритм для нахождения наибольшего общего делителя двух целых чисел (или общей меры двух отрезков). Алгоритм назван в честь греческого математика Евклида (III век до н. э.), который впервые описал его в VII и X книгах «Начал». Это один из старейших численных алгоритмов, используемых в наше время.

В самом простом случае алгоритм Евклида применяется к паре положительных целых чисел и формирует новую пару, которая состоит из меньшего числа и разницы между большим и меньшим числом. Процесс повторяется, пока числа не станут равными. Найденное число и есть наибольший общий делитель исходной пары. Евклид предложил алгоритм только для натуральных чисел и геометрических величин (длин, площадей, объёмов). Однако в XIX веке он был обобщён на другие типы математических объектов, включая целые числа Гаусса и полиномы от одной переменной. Это привело к появлению в современной общей алгебре такого понятия, как евклидово кольцо. Позже алгоритм Евклида был обобщён на другие математические структуры, такие как узлы и многомерные полиномы.

Для данного алгоритма существует множество теоретических и практических применений. В частности, он является основой для криптографического алгоритма с открытым ключом RSA, широко распространённого в электронной коммерции. Также алгоритм используется при решении линейных диофантовых уравнений, при построении непрерывных дробей, в методе Штурма. Алгоритм Евклида является основным инструментом для доказательства теорем в современной теории чисел, например таких как теорема Лагранжа о сумме четырёх квадратов и основная теорема арифметики.

# 4 Выполнение лабораторной работы

В ходе выполнения лабораторной работы было реализовано вариации 4 алгоритма по нахождению НОД.

Среди них:

* классический алгоритм по поиску НОД
* расширенный классический алгоритм выводящий НОД, а также коэффициенты Безу.
* бинарный алгоритм по поиску НОД
* расширенный бинарный алгоритм выводящий НОД, а также коэффициенты Безу.

Программный код представлен в качестве листинга в конце отчета.

# 5 Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: я ознакомился с алгоритмом Евклида, а так же мне удалось реализовать вариции этого алгоритма на языке программирования Python.

# 6 Листинг программы

def euclidean():  
 b = 1  
 a = 0  
 while(b>a):  
 a = int(input("a: "))  
 b = int(input("b: "))  
 if(b>a):  
 print("b cant be greater than a")  
   
   
 while b != 0:  
 t = b  
 b = a%b  
 a = t  
 print("НОД = ", a)  
 return a  
  
def euclidean\_binary():  
 b = 1  
 a = 0  
 while(b>a):  
 a = int(input("a: "))  
 b = int(input("b: "))  
 if(b>a):  
 print("b cant be greater than a")  
   
  
 g = 1  
 while((a % 2 == 0) | (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 = u/2  
 if(u>=v):  
 u = u-v  
 else:  
 v = v - u  
 d = g\*v  
 print("НОД = ",d)  
   
def euclidean\_extended():  
 b = 1  
 a = 0  
 while(b>a):  
 a = int(input("a: "))  
 b = int(input("b: "))  
 if(b>a):  
 print("b cant be greater than a")  
  
 x0 = 1  
 x1 = 0  
 y0 = 0  
 y1 = 1  
  
 while b != 0:  
 t = b  
 q = a // b  
 print(q)  
 b = a%b  
 a = t  
 t\_x = x1  
 x1 = x0 - q\*x1  
 x0 = t\_x  
 t\_y = y1  
 y1 = y0 - q\*y1  
 y0 = t\_y  
  
 print("НОД = ", a)  
 print("коэффициенты Безу: ", x0,y0)  
  
def euclidean\_binary\_extended():  
 b = 1  
 a = 0  
 while(b>a):  
 a = int(input("a: "))  
 b = int(input("b: "))  
 if(b>a):  
 print("b cant be greater than a")  
   
 g = 1  
 while((a % 2 == 0) | (b % 2 == 0)):  
 a = a/2  
 b = b/2  
 g = 2\*g  
 u = a   
 v = b  
 A = D = 1  
 B = C = 0  
 while(u != 0):  
 while(u % 2 == 0):  
 u = u/2  
 if((A % 2 == 0) & (B % 2 == 0)):  
 A = A/2  
 B = B/2  
 else:  
 A = (A+v)/ 2  
 B = (B-u)/ 2  
 while(v % 2 == 0):  
 v = u/2  
 if((C % 2 == 0) & (D % 2 == 0)):  
 A = A/2  
 B = B/2  
 else:  
 C = (C+v)/ 2  
 D = (D-u)/ 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  
   
 print("НОД = ",d)  
 print("коэффициенты Безу: ",C,D)  
  
# Вызовы функций  
  
euclidean()  
#euclidean\_extended()  
#euclidean\_binary()  
#euclidean\_binary\_extended()