Отчёт по лабораторной работе №4

Алгоритм Евклида

Гаглоев Олег Мелорович

Содержание

# 1 Цель работы

Изучение алгоритма Евклида нахождения НОД и его вариаций.

# 2 Теоретические сведения

## 2.1 Наибольший общий делитель

Наибольший общий делитель (НОД) – это число, которое делит без остатка два числа и делится само без остатка на любой другой делитель данных двух чисел. Проще говоря, это самое большое число, на которое можно без остатка разделить два числа, для которых ищется НОД.

## 2.2 Алгоритм Евклида

При работе с большими составными числами их разложение на простые множители, как правило, неизвестно. Но для многих прикладных задач теории чисел поиск разложения числа на множители является важной, часто встречающейся практической задачей. В теории чисел существует сравнительно быстрый способ вычисления НОД двух чисел, который называется алгоритмом Евклида.

Алгоритм Евклида

* Вход. Целые числа .
* Выход. НОД.

1. Положить , , .
2. Найти остаток от деления на .
3. Если , то положить . В противном случае положить и вернуться на шаг 2.
4. Результат: .

## 2.3 Бинарный алгоритм Евклида

Бинарный алгоритм Евклида вычисления НОД оказывается более быстрым при реализации этого алгоритма на компьютере, поскольку использует двоичное представление чисел а и b. Бинарный алгоритм Евклида основан на следующих свойствах наибольшего общего делителя (считаем, что 0 < b ≤ а):

* Вход. Целые числа .
* Выход. HOД.

1. Положить .
2. Пока оба числа и четные, выполнять до получения хотя бы одного нечетного значения или .
3. Положить .
4. Пока , выполнять следующие действия.

* Пока четное, полагать .
* Пока четное, полагать .
* При положить . В противном случае положить .

1. Положить .
2. Результат:

## 2.4 Расширенный алгоритм Евклида

Расширенный алгоритм Евклида находит наибольший общий делитель d чисел а и b и его линейное представление, т. е. целые числа x и у, для которых ах + by = d, и не требует «возврата», как в рассмотренном примере. Пусть d – НОД для a и b, т. е. d = (a, b), где a > b. Тогда существуют такие целые числа x и y, что d = ax +by. Иными словам, НОД двух чисел можно представить в виде линейной комбинации этих чисел с целыми коэффициентами

* Вход. Целые числа .
* Выход: НОД; такие целые числа , что .

1. Положить
2. Разделить с остатком на :
3. Если , то положить , , . В противном случае положить , , и вернуться на шаг 2.
4. Результат: .

# 3 Выполнение работы

## 3.1 Реализация алгоритмов на языке Python

## 3.2 Алгоритм Евклида

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

## 3.3 Бинарный Алгоритм Евклида

def Bin\_Euclide(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 int(d)

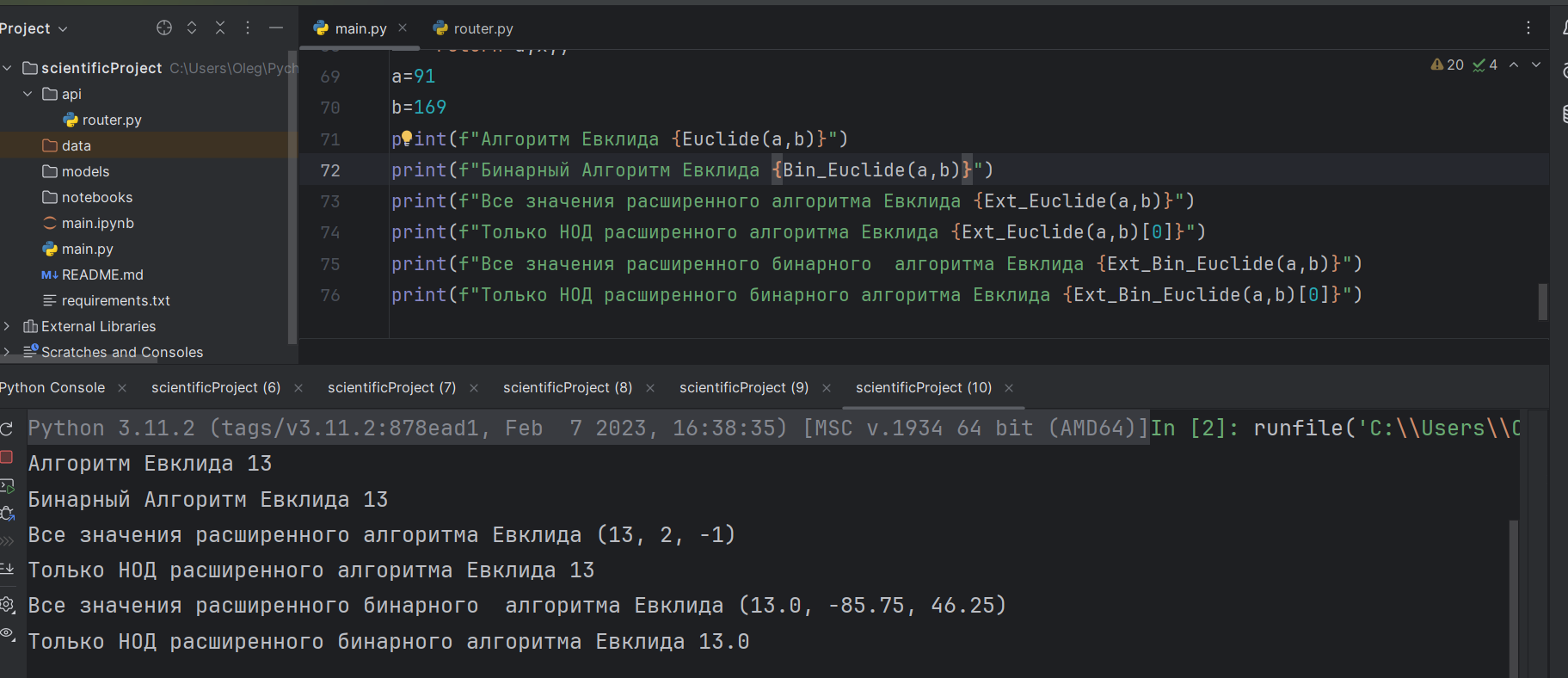
## 3.4 Расширенный Алгоритм Евклида

def Ext\_Euclide(a,b):  
 if a==0:  
 return b,0,1  
 else:  
 r,x,y=Ext\_Euclide(b%a,a)  
 return r,y-(b//a)\*x,x

## 3.5 Расширенный Бинарный Алгоритм Евклида

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

## 3.6 Контрольный пример



Работа алгоритмов

# 4 Выводы

Изучили алгоритм Евклида нахождения НОД.

# Список литературы

1. [ВЫЧИСЛЕНИЕ НАИБОЛЬШЕГО ОБЩЕГО ДЕЛИТЕЛЯ](http://ikit.edu.sfu-kras.ru/files/15/l2.pdf)
2. [В очередной раз о НОД](https://habr.com/ru/post/464949/)