Отчёт по лабораторной работе №4

Алгоритм Евклида

Гердт Ольга НФИмд-02-21"

Содержание"

# Цель работы

Изучение алгоритма Евклида нахождения НОД и его вариаций. Реализовать рассмотренные алгоритмы программно.

# Теоретические сведения

## 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.

## Алгоритм Евклида

**Алгоритм Евклида** – это алгоритм нахождения наибольшего общего делителя (НОД) пары целых чисел.

Алгоритм Евклида

Вход. Целые числа .

Выход. НОД.

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

## Бинарный алгоритм Евклида

Бинарный алгоритм Евклида — метод нахождения наибольшего общего делителя двух целых чисел. Возможно, алгоритм был известен еще в Китае 1-го века, но опубликован был лишь в 1967 году израильским физиком и программистом Джозефом Стайном. Он основан на использовании следующих свойств НОД:

* НОД() = 2 НОД(),
* НОД() = НОД(),
* НОД() = НОД()

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 .
5. Положить .
6. Результат:

## Расширенный алгоритм Евклида

В то время как "обычный" алгоритм Евклида просто находит наибольший общий делитель двух чисел и , расширенный алгоритм Евклида находит помимо НОД также коэффициенты и такие, что:  
НОД()

Т.е. он находит коэффициенты, с помощью которых НОД двух чисел выражается через сами эти числа.

Вход. Целые числа .

Выход: НОД; такие целые числа , что .

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

## Расширенный бинарный алгоритм Евклида

Вход. Целые числа:

Выход. = HOД().

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

* 4.1. Пока четное:
* 4.1.1. Положить
* 4.1.2. Если оба числа и четные,
  + то положить
  + иначе
* 4.2. Пока четное:
* 4.2.1. Положить
* 4.2.2. Если оба числа и четные,
  + то положить
  + иначе
* 4.3 При
  + положить
  + иначе

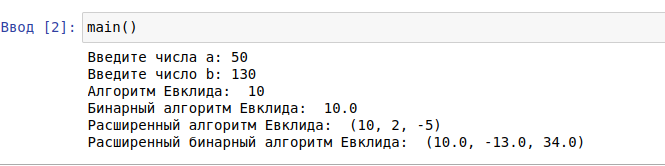
1. Положить
2. Результат:

# Выполнение работы

## Реализация алгоритмов на языке Python

# Алгоритм Евклида  
def evklid(a,b):  
 while a != 0 and b != 0:  
 if a >= b:  
 a %= b  
 else:  
 b %= a  
 return a or b  
  
# Бинарный алгоритм Евклида  
def binary\_evklid(a,b):  
 g = 1  
 while(a % 2 == 0 and b % 2 == 0):  
 a = a/2  
 b = b/2  
 g = 2\*g  
 u,v = a,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  
  
# Расширенный алгоритм Евклида  
def evklid\_extended(a, b):  
 if a == 0:  
 return (b, 0, 1)  
 else:  
 div, x, y = evklid\_extended(b % a, a)  
 return (div, y - (b // a) \* x, x)  
  
# Расширенный бинарный алгоритм Евклида  
def evklid\_binary\_extended(a, 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  
 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  
 return (d,x,y)  
  
def zapusk(a, b):  
 print("Алгоритм Евклида: ", evklid(a,b))  
 print("Бинарный алгоритм Евклида: ", binary\_evklid(a,b))  
 print("Расширенный алгоритм Евклида: ", evklid\_extended(a,b))  
 print("Расширенный бинарный алгоритм Евклида: ", evklid\_binary\_extended(a,b))  
   
def main():  
 a = int(input("Введите числа a: "))  
 b = int(input("Введите число b: "))  
 if a > b and b > 0:  
 zapusk(a, b)  
 elif a <= b and a > 0:  
 a, b = b, a   
 zapusk(a, b)  
 else:  
 print("Введены не коректные данные.")

## Контрольный пример



# Выводы

Изучили алгоритм Евклида, и его варианты. Реализовали алгоритмы програмно.

# Список литературы

1. [Алгоритм Евклида - нахождение наибольшего общего делителя](https://younglinux.info/algorithm/euclidean)
2. [Бинарный алгоритм вычисления НОД](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%9D%D0%9E%D0%94#cite_note-1)
3. [Расширенный алгоритм Евклида](https://spravochnick.ru/informatika/rasshirennyy_algoritm_evklida/)
4. [Расширенный алгоритм Евклида](https://aspektcenter.ru/rasshirennyy-algoritm-yevklida-s-tablitsey/)