

Алгоритмы Евклида для нахождения НОД.

Меньшов Иван Сергеевич

02 декабря, 2021, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Изучить алгоритмы для вычисления наибольшего общего делителя.

Выполнение лабораторной работы

Алгоритм Евклида

Для вычисления наибольшего общего делителя двух целых чисел применяется способ повторного деления с остатком, называемый алгоритмом Евклида.

Вход. Целые числа a, b ; $0 < b < a$. Выход. $d = \text{НОД}(a, b)$.

1. Положить $r_0 = a, r_1 = b, i = 1$
2. Найти остаток r_{i+1} от деления r_{i-1} на r_i
3. Если $r_{i+1} = 0$, то положить $d = r_i$. В противном случае положить $i = i+1$ и вернуться на шаг 2
4. Результат: d

Бинарный алгоритм Евклида

Бинарный алгоритм Евклида является более быстрым при реализации на компьютере, поскольку использует двоичное представление чисел a и b . Бинарный алгоритм Евклида основан на следующих свойствах наибольшего общего делителя (считаем, что $0 < b < a$):

1. если оба числа a и b четные, то $\text{НОД}(a,b) = 2 \text{НОД}(a/2,b/2)$
2. если число a - нечетное, число b — четное, то $\text{НОД}(a,b) = \text{НОД}(a, b/2)$
3. если оба числа a и b нечетные, $a > b$, то $\text{НОД}(a,b) = \text{НОД}(a - b, b)$
4. если $a = b$, то $\text{НОД}(a,b) = a$

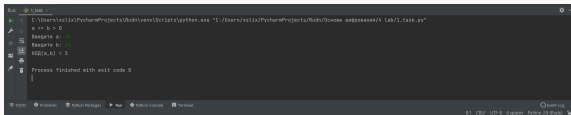
Расширенный алгоритм Евклида и Бинарный алгоритм Евклида

Данные алгоритмы Евклида находят, помимо $g = \text{НОД}(a, b)$ такие целые коэффициенты x и y , что:

$$ax + by = d$$

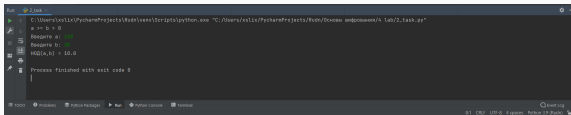
Заметим, что решений бесконечно много: имея решение (x, y) , можно x увеличить на b , а y уменьшить на a , и равенство при этом не изменится.

Контрольные примеры



```
C:\Users\vs1ix\PycharmProjects\Road\envs\Scripts\python.exe "C:/Users/vs1ix/PycharmProjects/Road/venv/venv/Scripts/5 Lab/2 task.py"
a = b = 5
Requests a: 5
Requests b: 5
HGD(a,b) = 5
Process finished with exit code 0
```

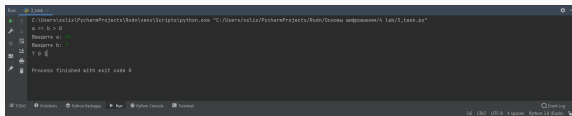
Figure 1: Алгоритм Евклида



```
C:\Users\vs1ix\PycharmProjects\Road\envs\Scripts\python.exe "C:/Users/vs1ix/PycharmProjects/Road/venv/venv/Scripts/5 Lab/2 task.py"
a = b = 5
Requests a: 5
Requests b: 10.0
HGD(a,b) = 10.0
Process finished with exit code 0
```

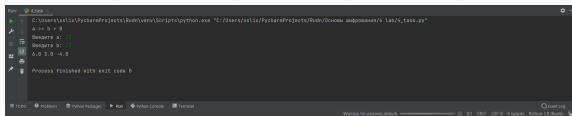
Figure 2: Бинарный алгоритм Евклида

Контрольные примеры



```
C:\Users\valia\PycharmProjects\Буду\venv\Scripts\python.exe "C:/Users/valia/PycharmProjects/Буду/Основы_математики/4_Lab/5_task.py"
a = 10, b = 8
Запрос: a: 10
Запрос: b: 8
7 0 5
Process finished with exit code 0
```

Figure 3: Расширенный алгоритм Евклида



```
C:\Users\valia\PycharmProjects\Буду\venv\Scripts\python.exe "C:/Users/valia/PycharmProjects/Буду/Основы_математики/4_Lab/5_task.py"
a = 10, b = 8
Запрос: a: 10
Запрос: b: 8
6.0 5.0 -4.0
Process finished with exit code 0
```

Figure 4: Расширенный бинарный алгоритм Евклида

Выводы

Мной были узчены алгоритмы для вычисления наибольшего общего делителя Евклида.