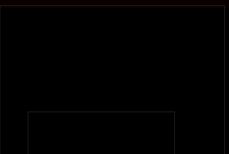


Лабораторная работа № 4

Алгоритм Евклида

Выполнил: Хамза Хуссен





Цели и задачи исследования

Цель работы:

Ознакомление с алгоритмом Евклида для вычисления НОД и изучение его ключевых модификаций.

Основные задачи:

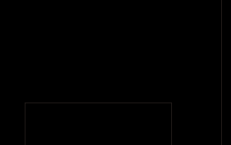
Изучить теоретические основы понятия НОД.

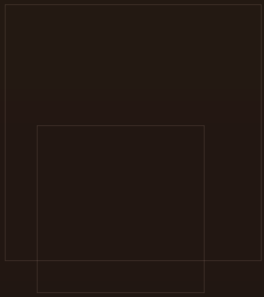
Разобрать классический алгоритм Евклида.

Изучить оптимизированный бинарный алгоритм.

Освоить расширенный алгоритм Евклида.

Реализовать изученные алгоритмы на языке Python.





Наибольший общий делитель (НОД)

Определение: Наибольшее натуральное число, которое делит два заданных целых числа без остатка.

Математическое обозначение: $\text{НОД}(a, b)$ или $\text{gcd}(a, b)$.

Области применения:

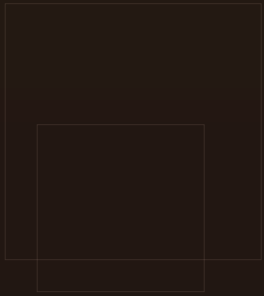
Теория чисел.

Криптография (например, RSA).

Упрощение дробей.

Алгоритмизация и программирование.

Классический алгоритм Евклида



Основное правило (лемма):
 $\text{НОД}(a, b) = \text{НОД}(b, a \bmod b)$, где $a \geq b > 0$.

Последовательность действий:

$r_0 = a, r_1 = b, i = 1$.

Найти остаток $r_{i+1} = r_{i-1} \bmod r_i$.

Если $r_{i+1} = 0$, то $\text{НОД} = r_i$.

Иначе увеличить i и вернуться к шагу 2.

Пример: $\text{НОД}(30, 18)$

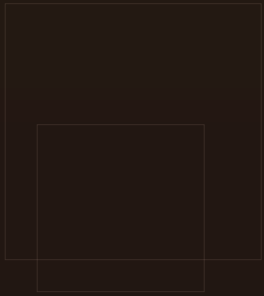
$$30 \% 18 = 12$$

$$18 \% 12 = 6$$

$$12 \% 6 = 0$$

Ответ: $\text{НОД} = 6$

Бинарный алгоритм Евклида



Идея: Оптимизация за счёт использования битовых операций (деление/умножение на 2).

Основные шаги:

Факторизация двойки: Пока a и b чётные, делим их на 2 и запоминаем множитель g .

Основной цикл:

Делим u и v на 2, пока они чётные.

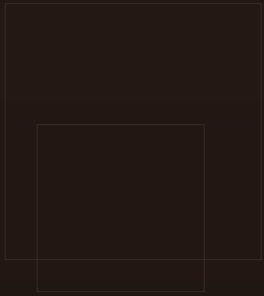
Вычитаем меньшее из большего: $u = u - v$ или $v = v - u$.

Повторяем, пока $u \neq 0$.

Результат: $\text{НОД} = g * v$.

Преимущество: Высокая скорость на компьютерах.

Расширенный алгоритм Евклида



Ключевое свойство:

Находит не только $d = \text{НОД}(a, b)$, но и целые коэффициенты x и y такие, что:

$$a \cdot x + b \cdot y = d$$

Этапы:

Инициализация: $(r_0, x_0, y_0) = (a, 1, 0)$, $(r_1, x_1, y_1) = (b, 0, 1)$.

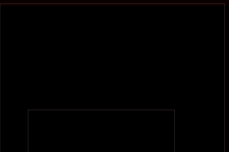
Пока остаток не 0:

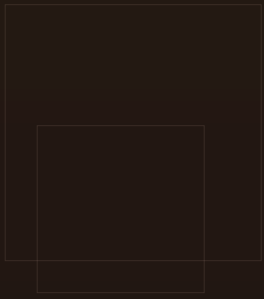
Найти частное $q = r_{i-1} // r_i$.

Вычислить новые значения: $r_{i+1} = r_{i-1} - q * r_i$, $x_{i+1} = x_{i-1} - q * x_i$, $y_{i+1} = y_{i-1} - q * y_i$.

Результат: $(d, x, y) = (r_i, x_i, y_i)$.

Применение: Решение линейных диофантовых уравнений, нахождение обратного элемента по модулю





Практическая реализация (Часть 1)

Практическая реализация (Часть 1)

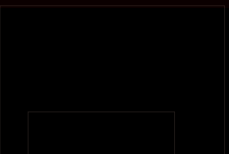
Ключевые функции на Python:

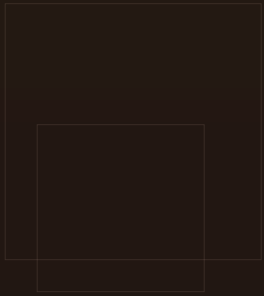
Классический алгоритм (euklid_simply): Последовательное применение $a \% b$.

Расширенный алгоритм (euklid_extended): Рекурсивное вычисление коэффициентов x и y .

Бинарный алгоритм (euklid_binary): Использование операций $// 2$ и вычитания.

Расширенный бинарный (euklid_bin_extended): Комбинация бинарного метода с поиском коэффициентов.





Практическая реализация (Часть 2)

Пример работы программы (фрагмент вывода):

Введите первое число (a): 30

Введите второе число (b): 18

1. Простой алгоритм Евклида:

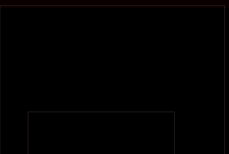
НОД(30, 18) = 6

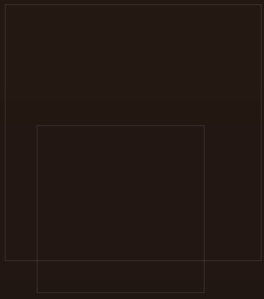
2. Расширенный алгоритм Евклида:

НОД(30, 18) = 6

Коэффициенты: $30 \cdot (-1) + 18 \cdot (2) = 6$

Проверка: $30 \cdot (-1) + 18 \cdot 2 = -30 + 36 = 6$





В ы в о д ы

В р е з у л ь т а т е р а б о т ы :

Освоен классический алгоритм Евклида — основа для вычисления НОД.

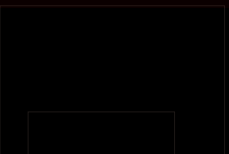
Изучены современные модификации:

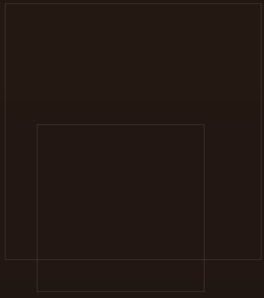
Бинарный алгоритм — эффективен для программной реализации.

Расширенный алгоритм — имеет фундаментальное значение в криптографии и теории чисел.

Получены практические навыки реализации этих алгоритмов на языке Python.

Подтверждена корректность работы алгоритмов на контрольных примерах.





С п а с и б о з а в н и м а н и е !

В о п р о с ы ?

