Лабораторная работа 4

Попов Дмитрий Павлович, НФИмд-01-23

Содержание

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра математического моделирования и искусственного интеллекта

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

дисциплина: Математические основы защиты информации и информационной безопасности

Преподователь: Кулябов Дмитрий Сергеевич

Студент: Попов Дмитрий Павлович

Группа: НФИмд-01-23

МОСКВА

2023 г.

# 1 Цель работы

Целью данной работы является освоить на практике алгоритмы вычисления НОД.[1]

# 2 Выполнение лабораторной работы

Требуется реализовать:

1. Алгоритм Евклида
2. Бинарный алгоритм Евклида
3. Расширенный алгоритм Евклида
4. Расширенный бинарный алгоритм Евклида

## 2.1 Алгоритм Евклида

Основные шаги:

1. Берёт два числа a и b, где a > b
2. Повторяет деление a на b, заменяя a значением b и b остатком от деления, пока b не станет равным 0.
3. Последнее ненулевое значение a будет НОД.

Реализация на Python предствлена на рисунке 1 fig. 1.

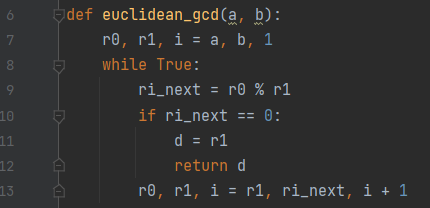


Figure 1: nod1

## 2.2 Бинарный алгоритм Евклида

Основные шаги:

1. Если оба числа четные, делит оба числа на 2 и удваивает итоговый НОД
2. Если только одно из чисел четное, делит только его на 2.
3. Из большего числа вычитает меньшее.
4. Повторяет процесс, пока числа не станут равными. Это число становится НОД, умноженным на полученный ранее множитель.

Реализация на Python предствлена на рисунке 2 fig. 2.

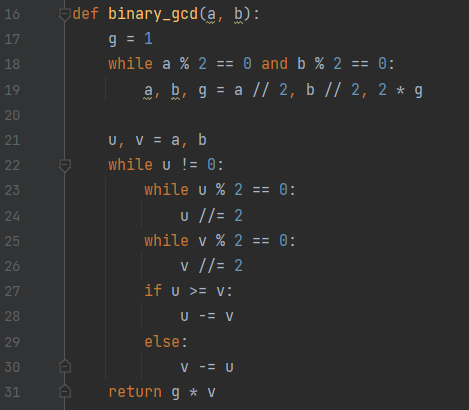


Figure 2: nod2

## 2.3 Расширенный алгоритм Евклида

Основные шаги:

1. Кроме нахождения НОД, алгоритм находит такие числа x и y, что ax+by=НОД(a,b).
2. Начинается с базовых коэф.: x0 = 1, y0 = 0 (для а) и x1 = 0, y1 = 1 (для b).
3. При каждом шаге обновляются значения коэффициентов, используя остаток и частное от деления.

Реализация на Python предствлена на рисунке 3 fig. 3.

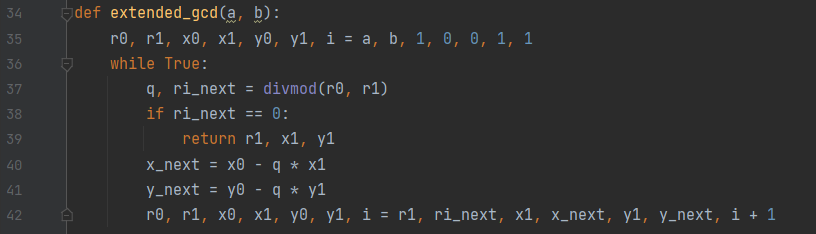


Figure 3: nod3

## 2.4 Расширенный бинарный алгоритм Евклида

Основные шаги:

1. Как и обычный бинарный алгоритм, но также отслеживает коэффициенты x и y.
2. Когда числа делятся на 2, коэффициенты корректируются соответствующим образом.
3. Когда из одного числа вычитается другое, соответствующие коэффициенты также вычитаются.

Реализация на Python предствлена на рисунке 4 fig. 4.

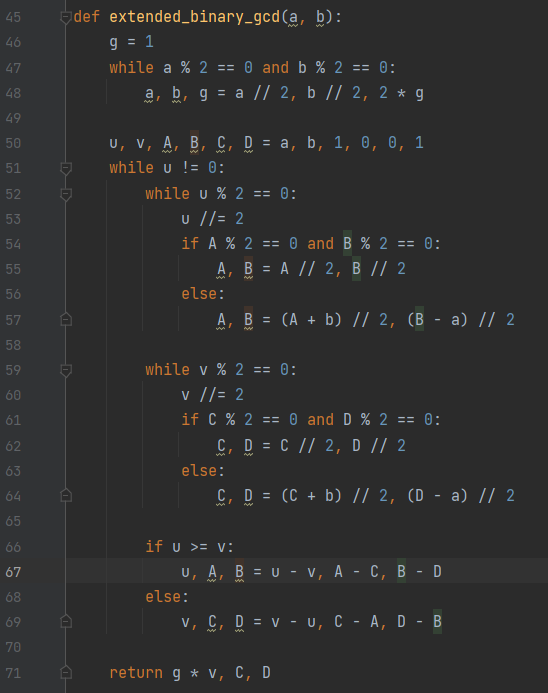


Figure 4: nod4

## 2.5 Результат работы программы

Выходные значения программы fig. 5.

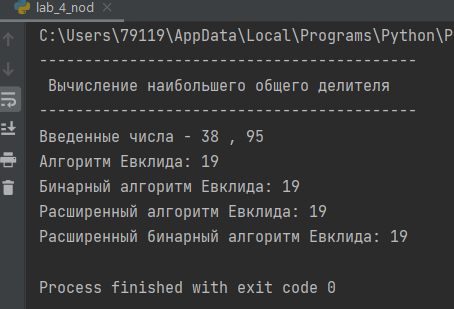


Figure 5: out

# 3 Выводы

В результате выполнения работы я освоил на практике применение алгоритмов нахождения НОД.

# 4 Список литературы

1. Методические материалы курса