Отчет по лабораторной работе №6

Разложение чисел на множители

Бурдина Ксения Павловна

23 ноября 2023

Содержание

# 1 Цель работы

Целью данной работы является освоение *p-метода Полларда*, который является одним из алгоритмом разложения составного числа на множители.

# 2 Задание

1. Изучить алгоритм разложения чисел на множители.
2. Реализовать представленный алгоритм и разложить на множители заданное число.

# 3 Теоретическое введение

Задача разложения на множители - одна из первых задач, использованных для построения криптосистем с открытым ключом.

*Задача разложения составного числа на множители* формулируется следующим образом: для данного положительного целого числа *n* найти его каноническое разложение , где - попарно различные простые числа, .

На практике не обязательно находить каноническое разложение числа *n*. Достаточно найти его разложение на два *нетривиальных сомножителя*: . Далее будем понимать задачу разложения именно в этом смысле.

*p-Метод Полларда*. Пусть *n* - нечетное составное число, и - случайное отображение. обладающее сжимающими свойствами. например. ( ). Основная идея метода состоит в следующем. Выбираем случайный элемент и строим последовательность , определяемую рекуррентным соотношением

где , до тех пор, пока не найдем такие числа , что и . Поскольку множество конечно, такие индексы существуют (последовательность “зацикливается”) [[2]](https://esystem.rudn.ru/pluginfile.php/2089897/mod_folder/content/0/mathsec_lection12-message-integrity-authentication.pdf?forcedownload=1). Последовательность будет состоять из “хваста” длины и цикла той же длины.

## 3.1 Алгоритм, реализующий р-метод Полларда

*Вход*. Число , начальное значение , функция , обладающая сжимающими свойствами.

*Выход*. Нетривиальный делитель числа .

* положить ,
* вычислить ( ), ( )
* найти
* если , то положить и результат: . При результат: “Делитель не найден”; при вернуться на шаг 2

**Пример** [[1]](https://intuit.ru/studies/courses/552/408/lecture/9350). Найти р-методом Полларда нетривиальный делитель числа . Положим и ( ). Работа алгоритма иллюстрируется следующей таблицей:

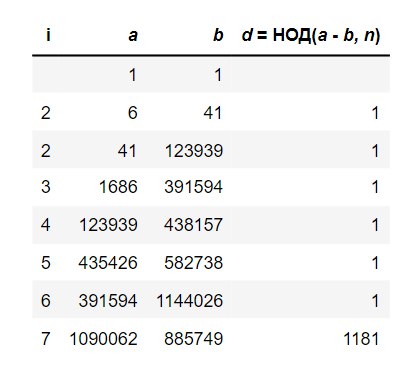


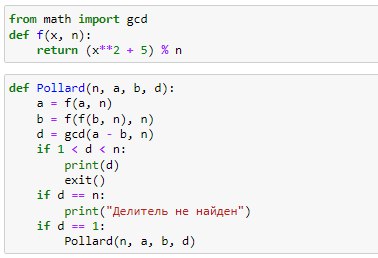
Схема работы алгоритма

Таким образом, является нетривиальным делителем числа .

# 4 Ход выполнения лабораторной работы

Для реализации рассмотренного алгоритма разложения чисел на множители будем использовать среду JupyterLab. Выполним необходимую задачу.

1. Подключим необходимые для работы библиотеки, в частности, библиотеку нахождения НОД.
2. Пропишем функцию , обладающую сжимающими свойствами.
3. Запишем алгоритм, реализующий *р-метод Полларда*, с помощью следующей функции:



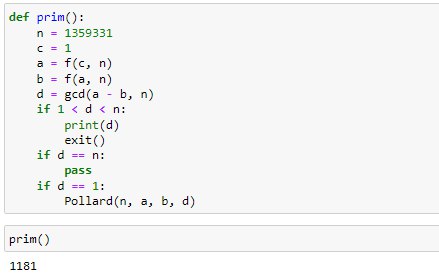
Реализация метода Полларда

Здесь на вход подается целое число , начальные значения и , вычисляемые через вышеописанную функцию. Необходимо выполнить следующее:

* положить ,
* вычислить ( ), ( )
* найти
* если , то положить и результат: . При результат: “Делитель не найден”; при вернуться на шаг 2

По итогу при вызове функции мы получим нетривиальный делитель числа .

1. Проверим корректность работы алгоритма для заданных сведений. Для этого запишем условие примера с помощью следующей функции:



Пример алгоритма

При вызове данной функции видим, что получаем то же число, что было описано в примере. То есть является нетривиальным делителем числа .

# 5 Листинг программы

from math import gcd  
def f(x, n):  
 return (x\*\*2 + 5) % n  
  
def Pollard(n, a, b, d):  
 a = f(a, n)  
 b = f(f(b, n), n)  
 d = gcd(a - b, n)  
 if 1 < d < n:  
 print(d)  
 exit()  
 if d == n:  
 print("Делитель не найден")  
 if d == 1:  
 Pollard(n, a, b, d)  
  
def prim():  
 n = 1359331  
 c = 1  
 a = f(c, n)  
 b = f(a, n)  
 d = gcd(a - b, n)  
 if 1 < d < n:  
 print(d)  
 exit()  
 if d == n:  
 pass  
 if d == 1:  
 Pollard(n, a, b, d)  
  
prim()

# 6 Выводы

В ходе работы мы изучили и реализовали вероятностные алгоритмы проверки чисел на простоту.

# 7 Список литературы

1. Фороузан Б. А. Криптография и безопасность сетей. - М.: Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2010. - 784 с. [[1]](https://intuit.ru/studies/courses/552/408/lecture/9350)
2. Методические материалы курса [[2]](https://esystem.rudn.ru/pluginfile.php/2089897/mod_folder/content/0/mathsec_lection12-message-integrity-authentication.pdf?forcedownload=1)