Отчёт по лабораторной работе №7

Дискретное логарифмирование в конечном поле

Хитяев Евгений Анатольевич НПМмд-02-21

Содержание

# 1 Цель работы

Изучение задачи дискретного логарифмирования.

# 2 Теоретические сведения

Пусть в некоторой конечной мультипликативной абелевой группе задано уравнение

Решение задачи дискретного логарифмирования состоит в нахождении некоторого целого неотрицательного числа , удовлетворяющего уравнению. Если оно разрешимо, у него должно быть хотя бы одно натуральное решение, не превышающее порядок группы. Это сразу даёт грубую оценку сложности алгоритма поиска решений сверху — алгоритм полного перебора нашёл бы решение за число шагов не выше порядка данной группы.

Чаще всего рассматривается случай, когда группа является циклической, порождённой элементом . В этом случае уравнение всегда имеет решение. В случае же произвольной группы вопрос о разрешимости задачи дискретного логарифмирования, то есть вопрос о существовании решений уравнения , требует отдельного рассмотрения.

## 2.1 p-алгоритм Полларда

* Вход. Простое число , число порядка по модулю , целое число б ; отображение , обладающее сжимающими свойствами и сохраняющее вычислимость логарифма.
* Выход. показатель , для которого , если такой показатель существует.

1. Выбрать произвольные целые числа и положить
2. Выполнять $c=f(c)(mod p), d=f(f(d))(mod p), вычисляя при этом логарифмы для и как линейные функции от по модулю , до получения равенства
3. Приняв логарифмы для и , вычислить логарифм решением сравнения по модулю . Результат или “Решения нет”.

# 3 Выполнение работы

## 3.1 Реализация алгоритмов на языке Python

from math import gcd  
  
ag = 1  
bg = 1  
  
def f(x, n):  
 return (x\*x+5)%n  
  
def method(n, a, b, d):  
 a = f(a, n)%n  
 b = f(f(b,n), n)%n  
 d = gcd(a-b, n)  
 if 1 < d < n:  
 p = d  
 print(p)  
 exit()  
 if d == n:  
 print("Делитель не найден")  
 if d == 1:  
 global ag  
 ag = b  
 method(n, a, b, d)  
  
def main():  
 n = 1359331  
 c = 1  
 a = c  
 b = c  
 a = f(a, n)%n  
 b = f(a, n)%n  
 d = gcd(a-b, n)  
 if 1 < d < n:  
 p = d  
 print(p)  
 exit()  
 if d == n:  
 pass  
 if d == 1:  
 method(n, a, b, d)  
  
main()

## 3.2 Контрольный пример

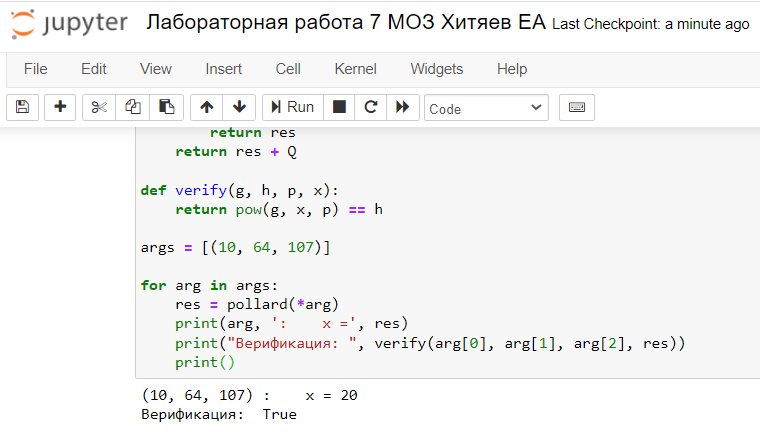


Figure 1: Пример работы алгоритма

Таким образом, число 1181 является нетривиальным делителем числа 1359331.

# 4 Выводы

В ходе выполнения работы мне удалось изучить задачу разложения на множители и p-алгоритм Полларда, а также реализовать данный алгоритм программно на языке Python.

# Список литературы

1. [Алгоритмы тестирования на простоту и факторизации](https://habr.com/ru/post/521876/)
2. [P-метод Полларда](https://ru.bmstu.wiki/P-метод_Полларда)