Отчёт по лабораторной работе №6

Разложение чисел на множители

Волкова Дарья Александровна НПМмд-02-21

Содержание

[Цель работы 1](#_Toc95131906)

[Теоретические сведения 1](#_Toc95131907)

[p-алгоритм Полларда 2](#_Toc95131908)

[Выполнение работы 2](#_Toc95131909)

[Реализация алгоритмов на языке Python 2](#_Toc95131910)

[Контрольный пример 4](#_Toc95131911)

[Выводы 4](#_Toc95131912)

[Список литературы 5](#_Toc95131913)

# Цель работы

Изучение задачи разложения чисел на множители, изучение p-алгоритма Полларда, а также его прогрммная реализация.

# Теоретические сведения

Разложение на множители — предмет непрерывного исследования в прошлом; и такие же исследования, вероятно, продолжатся в будущем. Разложение на множители играет очень важную роль в безопасности некоторых криптосистем с открытым ключом.

Согласно основной теореме арифметики любое положительное целое число больше единицы может быть уникально записано в следующей главной форме разложения на множители, где p\_1, p\_2, …, p\_k — простые числа и e\_1, e\_2, …, e\_k — положительные целые числа.

Поиск эффективных алгоритмов для разложения на множители больших составных чисел ведется давно. К сожалению, совершенный алгоритм для этого пока не найден. Хотя есть несколько алгоритмов, которые могут разложить число на множители, ни один не способен провести разложение достаточно больших чисел в разумное время. Позже мы увидим, что это хорошо для криптографии, потому что современные криптографические системы полагаются на этот факт. В этой секции мы даем несколько простых алгоритмов, которые проводят разложение составного числа. Цель состоит в том, чтобы сделать процесс разложения на множители менее трудоёмким.

В 1974 г. Джон Поллард разработал метод, который находит разложение числа на простые числа. Метод основан на условии, что p – 1 не имеет сомножителя, большего, чем заранее определенное значение B, называемое границей.

## p-алгоритм Полларда

Вход. Число n, начальное значение c, функция f, обладающая сжимающими свойствами.

Выход. Нетривиальный делитель числа n.

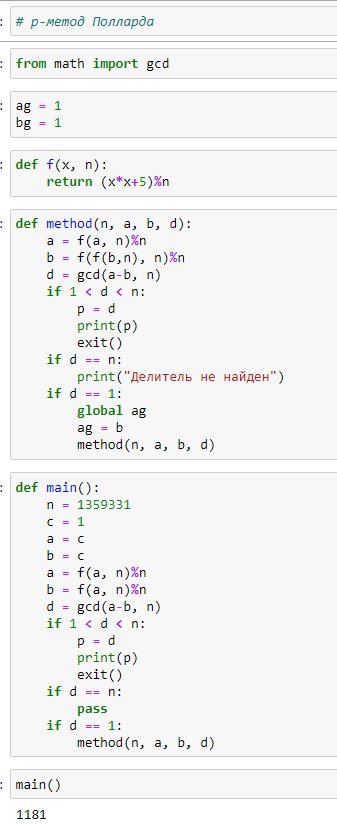
1. Положить a=c, b=c.
2. Вычислить a=f(a)(mod n), b=f(b)(mod n).
3. Найти d = НОД(a-b, n).
4. Если 1<d<n, то положить p=d и результат: p. При d=n результат: “Делитель не найден”. При d=1 вернуться на шаг 2.

# Выполнение работы

## Реализация алгоритмов на языке Python

# р-метод Полларда  
  
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()

## Контрольный пример



Пример работы алгоритма

Получили, что число 1181 является нетривиальным делителем числа 1359331.

# Выводы

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

# Список литературы

1. [Разложение на множители (факторизация)](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A0%D0%B0%D0%B7%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BD%D0%B0_%D0%BC%D0%BD%D0%BE%D0%B6%D0%B8%D1%82%D0%B5%D0%BB%D0%B8_(%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F))
2. [Факторизация чисел и методы решета. Часть I](https://habr.com/ru/post/521876/)