Отчёт по лабораторной работе №6

Разложение чисел на множители

Гаглоев Олег Мелорович

Содержание

# 1 Цель работы

Изучение задачи разложения на множители, изучение p-алгоритма Поллрада.

# 2 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 — простые числа и — 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 не имеет сомножителя, большего, чем заранее определенное значение , называемое границей. Алгоритм Полларда показывает, что в этом случае

Сложность. Заметим, что этот метод требует сделать операций возведения в степень . Есть быстрый алгоритм возведения в степень, который выполняет это за операций. Метод также использует вычисления НОД, который требует операций. Мы можем сказать, что сложность — так или иначе больше, чем или , где — число битов в . Другая проблема – этот алгоритм может заканчиваться сигналом об ошибке. Вероятность успеха очень мала, если имеет значение, не очень близкое к величине .

## 2.1 p-алгоритм Поллрада

* Вход. Число , начальное значение , функция , обладающая сжимающими свойствами.
* Выход. Нетривиальный делитель числа .

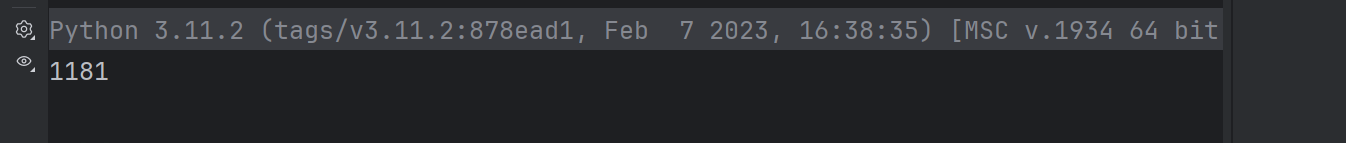
1. Положить
2. Вычислить
3. Найти
4. Если , то положить и результат: . При результат: ДЕЛИТЕЛЬ НЕ НАЙДЕН. При вернуться на шаг 2.

# 3 Выполнение работы

## 3.1 Реализация алгоритма на языке Python

def f(x,n):  
 return (x\*x+5)%n  
def Ext\_Euclide(a,b):  
 if a==0:  
 return b,0,1  
 else:  
 r,x,y=Ext\_Euclide(b%a,a)  
 return r,y-(b//a)\*x,x  
  
def Polard(n:int,c:int,fn):  
 a=c  
 b=c  
 while True :  
 a=fn(a,n)  
 b=fn(fn(b,n),n)  
 d=Ext\_Euclide(a-b,n)[0]  
 if 1<d<n:  
 return d  
 if d==n:  
 return -1  
 if d==1:  
 continue  
  
if \_\_name\_\_=="\_\_main\_\_":  
 n=1359331  
 c=1  
 print(Polard(n,c,f))

## 3.2 Контрольный пример



Работа алгоритма

# 4 Выводы

Изучили задачу разложения на множители и p-алгоритм Поллрада.

# 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)