Отчёт по лабораторной работе №6

Разложение чисел на множители

Волкова Дарья Александровна НПМмд-02-21

Содержание

Цель работы

Изучение задачи разложения чисел на множители, изучение ρ-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 p 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

```
# p-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

```
: # p-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()
```

```
1181
```

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)
2. Факторизация чисел и методы решета. Часть I