

Цели и задачи

Цель лабораторной работы

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

Выполнение лабораторной работы

Задача разложения на простые множители

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.

p -алгоритм Поллрада

- Вход. Число n , начальное значение c , функция f , обладающая сжимающими свойствами.
 - Выход. Нетривиальный делитель числа n .
- Положить $a=c$, $b=c$
 - Вычислить $a=f(a) \pmod n$, $b=f(b) \pmod n$
 - Найти $d = \text{GCD}(a-b, n)$
 - 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.

Оценка сложности

Сложность. Заметим, что этот метод требует сделать $B - 1$ операций возведения в степень $a = a^e \pmod n$. Есть быстрый алгоритм возведения в степень, который выполняет это за $2 \cdot \log_2 B$ операций. Метод также использует вычисления НОД, который требует n^3 операций. Мы можем сказать, что сложность — так или иначе больше, чем $O(B)$ или $O(2^n)$, где n_b — число битов в B . Другая проблема — этот алгоритм может заканчиваться сигналом об ошибке. Вероятность успеха очень мала, если B имеет значение, не очень близкое к величине \sqrt{n} .

Пример работы алгоритма

```
from math import gcd

def f(x, n):
    return (x*x+5)%n

def fu(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("not found")
    if d == 1:
        fu(n, a, b, d)

def main():
    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:
        fu(n, a, b, d)
```

Выводы

```
] : main()
```

```
1181
```

Результаты выполнения лабораторной работы

Изучили задачу разложения на множители и р-алгоритм Поллрада.