

# Разложение чисел на множители

---

Адьяту Ибрайма Коллаволе Топе НФИмд 01-22

22 ноября, 2022, Москва, Россия

Российский Университет Дружбы Народов

# Цели и задачи

---

## Цель лабораторной работы

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

# **Выполнение лабораторной работы**

---

## Задача разложения на простые множители

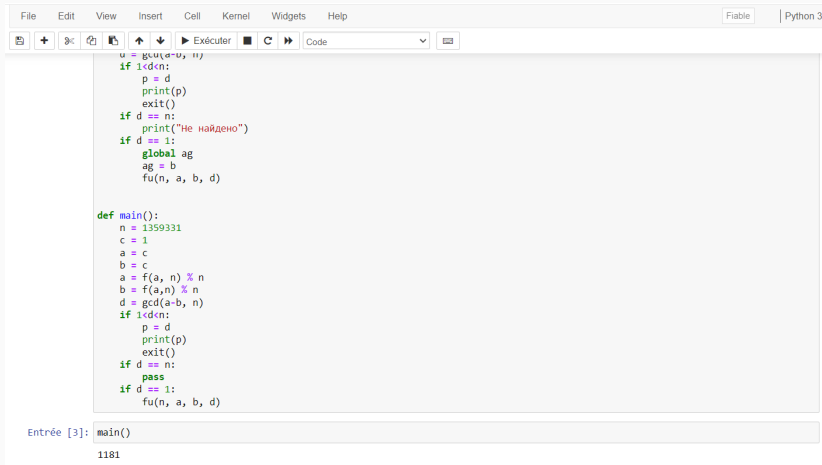
Разложение на множители — предмет непрерывного исследования в прошлом; и такие же исследования, вероятно, продолжатся в будущем. Разложение на множители играет очень важную роль в безопасности некоторых криптосистем с открытым ключом.

## р-алгоритм Поллрада

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

Сложность. Заметим, что этот метод требует сделать  $B-1$  операций возведения в степень  $a = a^e \bmod n$ . Есть быстрый алгоритм возведения в степень, который выполняет это за  $2 * \log_2 B$  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



The screenshot shows a Jupyter Notebook interface with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, execution, and code management. The main area contains a code cell with the following Python code:

```
u = gcd(a-b, n)
if 1<d<n:
    p = d
    print(p)
    exit()
if d == n:
    print("Не найдено")
if d == 1:
    global ag
    ag = b
    fu(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:
        fu(n, a, b, d)
```

Below the code cell, the input prompt "Entrée [3]:" is followed by the function call "main()". The output of the execution is the number "1181".

Figure 1: Работа алгоритма



## **Выводы**

---

Изучили задачу разложения на множители и р-алгоритм Поллрада.