

# Разложение чисел на множители

---

Мадаманов Аллаберды

7 февраля, 2024, Москва, Россия

Российский Университет Дружбы Народов

# Цели и задачи

---

## Цель лабораторной работы

Изучение задачи дискретного логарифмирования.

# **Выполнение лабораторной работы**

---

# Задача дискретного логарифмирования

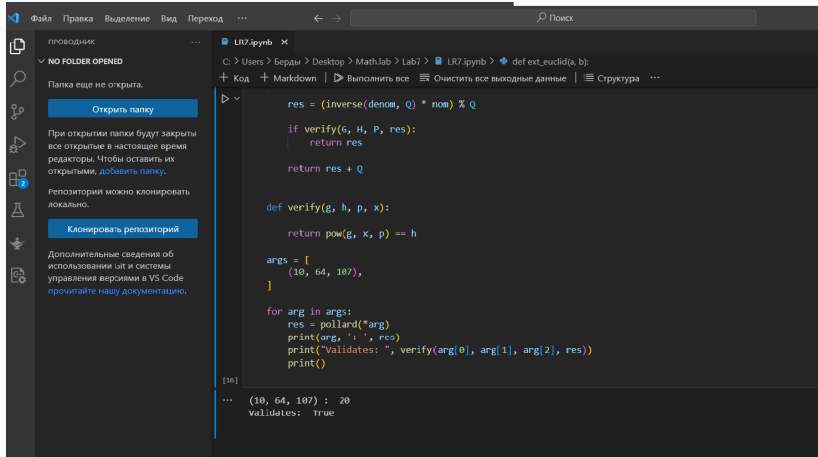
Решение задачи дискретного логарифмирования состоит в нахождении некоторого целого неотрицательного числа  $x$ , удовлетворяющего уравнению. Если оно разрешимо, у него должно быть хотя бы одно натуральное решение, не превышающее порядок группы.

## p-алгоритм Поллрада

- Вход. Простое число  $p$ , число  $a$  порядка  $r$  по модулю  $p$ , целое число  $b$   $1 < b < p$ ; отображение  $f$ , обладающее сжимающими свойствами и сохраняющее вычислимость логарифма.
  - Выход. показатель  $x$ , для которого  $a^x = b \pmod{p}$ , если такой показатель существует.
1. Выбрать произвольные целые числа  $u, v$  и положить  $c = a^u b^v \pmod{p}$ ,  $d = c$
  2. Выполнять  $c = f(c) \pmod{p}$ ,  $d = f(f(d)) \pmod{p}$ , вычисляя при этом логарифмы для  $c$  и  $d$  как линейные функции от  $x$  по модулю  $r$ , до получения равенства  $c = d \pmod{p}$
  3. Приняв логарифмы для  $c$  и  $d$ , вычислить логарифм  $x$  решением сравнения по модулю  $r$ . Результат  $x$  или  
РЕШЕНИЯ НЕТ

Алгоритм полного перебора нашёл бы решение за число шагов не выше порядка данной группы.

# Пример работы алгоритма



The screenshot shows the Visual Studio Code interface with a Jupyter notebook open. The left sidebar contains the Explorer view with a message 'NO FOLDER OPENED' and buttons to 'Открыть папку' (Open folder) and 'Клонировать репозиторий' (Clone repository). The main editor area displays the following Python code:

```
C: > Users > Берды > Desktop > Mathlab > Lab7 > LR7.ipynb > def ext_euclid(a, b):
+ Код + Markdown | ▶ Выполнить все | Очистить все выходные данные | Структура ...
def ext_euclid(a, b):
    res = (inverse(denom, Q) * nom) % Q

    if verify(6, H, P, res):
        return res

    return res + Q

def verify(g, h, p, x):
    return pow(g, x, p) == h

args = [
    (10, 64, 107),
]

for arg in args:
    res = pollard(*arg)
    print(arg, ' : ', res)
    print("Validates: ", verify(arg[0], arg[1], arg[2], res))
    print()

[16]
... (10, 64, 107) : 20
Validates: True
```

Рис. 1: Работа алгоритма



## **Выводы**

---

В данной лабораторной работе мы изучили задачу дискретного логарифмирования.