

Разложение чисел на множители

Миличевич Александра

15 Февраля, 2025, Москва, Россия

Российский Университет Дружбы Народов

Цели и задачи

Цель лабораторной работы

Ознакомление с r -методом Полларда

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

- Функция ``pollard_rho_function(x, n)`

Как работает:

1. ****Обновление значений:**** Вычисляются новые значения ``a_val`` и ``b_val`` путем итеративного применения функции ``pollard_rho_function``. Значение ``b_val`` обновляется дважды за итерацию.
2. ****Вычисление НОД:**** Вычисляется наибольший общий делитель (НОД) между разностью ``a_val`` и ``b_val`` и исходным числом ``number`` с помощью функции ``gcd`` из модуля ``math``.

3. **Проверка делителя:**

* Если ``divisor`` находится между 1 и ``number`` ($1 < \text{divisor} < \text{number}$), значит, найден нетривиальный делитель. Функция выводит этот делитель и завершает выполнение программы.

* Если ``divisor`` равен ``number``, то это означает неудачу, и функция выводит сообщение об этом.

* Если ``divisor`` равен 1, это означает, что на текущей итерации делитель не найден, и функция продолжает свою работу рекурсивно.

Оценка сложности

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} .

4. ****Использование глобальной переменной:****

- * Перед рекурсивным вызовом, текущее значение ``b_val`` сохраняется в глобальной переменной ``global_b``, что позволяет отслеживать значение ``b_val`` между рекурсивными вызовами.

- * ``global_b`` объявляется как глобальная переменная внутри функции с помощью ключевого слова ``global``.

Пример работы алгоритма

```
from math import gcd

# Глобальные переменные для отслеживания значений
global_a = 1
global_b = 1

def pollard_rho_function(x, n):
    """
    Функция  $f(x) = (x^2 + 5) \% n$ , используемая в р-методе Полларда.

    Args:
        x (int): Входное число.
        n (int): Модуль.

    Returns:
        int: Результат  $(x^2 + 5) \% n$ .
    """
    return (x*x+5)%n

def pollard_rho_recursive(number, a_val, b_val, divisor):
    """
    Рекурсивная функция для реализации р-метода Полларда.

    Args:
        number (int): Число, для которого ищется нетривиальный делитель.
        a_val (int): Текущее значение 'a'.
        b_val (int): Текущее значение 'b'.
        divisor (int): Текущий наибольший общий делитель.
    """
    # Вычисляем следующее значение 'a'
    a_val = pollard_rho_function(a_val, number) % number
    # Вычисляем следующее значение 'b' (дважды применяем функцию)
    b_val = pollard_rho_function(pollard_rho_function(b_val, number), number) % number
    # Вычисляем новый наибольший общий делитель
    divisor = gcd(a_val - b_val, number)

    # Если  $1 < \text{divisor} < \text{number}$ , мы нашли нетривиальный делитель, выводим его и завершаем работу.
    if 1 < divisor < number:
        print(divisor)
        exit()

    # Если divisor равен number, это означает неудачу, и мы ничего не выводим
    if divisor == number:
        print("Делитель не найден")
        return

    # Если divisor равен 1, продолжаем рекурсивно с новыми значениями
    if divisor == 1:
        global global_b # Используем глобальную переменную для сохранения значения b_val
        global_b = b_val # Сохраняем текущее значение b_val в глобальную переменную.
        pollard_rho_recursive(number, a_val, b_val, divisor)
```

Рис. 1: Pollard

Выводы

Результаты выполнения лабораторной работы

Изучили задачу разложения на множители и р-алгоритм Поллрада.