Отчёт по лабораторной работе №6. Разложение чисел на множители

Дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Аронова Юлия Вадимовна, 1032212303
Группа: НФИмд-01-21
Преподаватель: д-р.ф.-м.н., проф. Кулябов Дмитрий Сергеевич

17 декабря, 2021, Москва

Цели и задачи работы

Целью данной лабораторной работы является краткое ознакомление с ρ-методом Полларда для нахождения нетривиального делителя целого числа, а также его последующая программная реализация.

Задачи: Рассмотреть и реализовать на языке программирования Python ρ-метод Полларда для нахождения нетривиального делителя целого числа.

Теоретическое введение

Факторизация чисел

Факторизацией целого числа называется его разложение в произведение простых сомножителей. Такое разложение, согласно основной теореме арифметики, всегда существует и является единственным (с точностью до порядка следования множителей).

Мы будем ограничиваться поиском разложения на два *нетривиальных* множителя: $n = ab, 1 < a \le b < n$.

ρ-метод Полларда (1 / 3)

Этот метод был разработан Джоном Поллардом в 1975 г.
Пусть $n \in \mathbb{N}$ – число, которое следует разложить.

1 шаг: Выбрать отображение $f : \mathbb{Z}_n \to \mathbb{Z}_n$. Обычно $f(x)$ – многочлен степени большей или равной 2, например, $f(x) = x^2 + 1$.

2 шаг: Случайно выбрать $x_0 \in \mathbb{Z}_n$ и вычислять члены рекуррентной последовательности $x_0, x_1, x_2, \ldots : x_i \equiv f(x_{i-1}) \pmod{n}$.

3 шаг: Для некоторых номеров j, k проверять условие $1 < \text{НОД}(x_j - x_k, n) < n$ до тех пор, пока не будет найден делитель числа n.

ρ-метод Полларда (2 / 3)

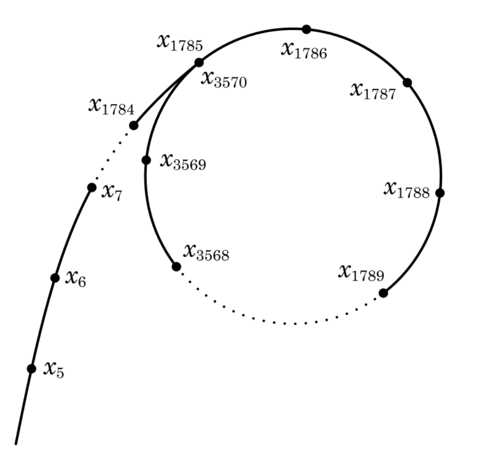

Figure 1: Зацикливание числовой последовательности, получаемой методом ρ-методом Полларда

ρ-метод Полларда (3 / 3)

Алгоритм 1. Алгоритм, реализующий ρ-метод Полларда

Вход. Число n, начальное значение c, функция f, обладающая сжимающими свойствами.

Выход. Нетривиальный делитель числа n.

1. Положить $a \leftarrow c, b \leftarrow c$.
2. Вычислить $a \leftarrow f(a) \pmod n, b \leftarrow f(f(b)) \pmod n$.
3. Найти $d \leftarrow$ НОД$(a - b, n)$.
4. При $1 < d < n$ положить $p \leftarrow d$ и результат: d. При $d = n$ результат: "Делитель не найден". При $d = 1$ вернуться на шаг 2.

Figure 2: Алгоритм, реализующий ρ-метод Полларда

Ход выполнения и результаты

Реализация (1 / 2)

```python
def euclidean_algorithm(a, b):
    """
    Находит НОД чисел a и b с помощью алгоритма Евклида
    """
    (a, b) = (abs(int(a)), abs(int(b)))
    if b > a:
        (a, b) = (b, a)
    r = [a, b]
    while r[1] != 0:
        (r[0], r[1]) = (r[1], r[0] % r[1])
    return r[0]
```

Реализация (2 / 2)

```
def pollard_rho_method(n, f, c = 1):
    a = c; b = c # шаг 1
    while True:
        x = a            #
        a = eval(f) % n  #
        x = b            #  шаг 2
        x = eval(f)      #
        b = eval(f) % n  #
        d = euclidean_algorithm(abs(a - b), n) # шаг 3
        if d > 1 and d < n:                #
            return d                       #
        if d == n:                         # шаг 4
            print("Делитель не найден")    #
            return 0                       #
```

```
print(pollard_rho_method(8051, "x ** 2 + 1"))
print(pollard_rho_method(8051, "x ** 2 + 3"))

print(pollard_rho_method(1359331, "x ** 2 + 5"))

print(pollard_rho_method(13562997737, "x ** 2 + 5"))
print(pollard_rho_method(13562997737, "x ** 2 + 1"))
```

[3] ✓ 0.4s

```
97
83
1181
89
419
```

Figure 3: Примеры нахождения нетривиальных делителей чисел посредством программной реализации ρ-метода Полларда

Заключение

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: было проведено краткое знакомство с алгоритмом, реализующим ρ-метод Полларда для нахождения нетривиального делителя целого числа, после чего алгоритм был успешно реализован на языке программирования **Python**.

Спасибо за внимание