РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Математические основы защиты информации и информационной безопасности

Студент: Пиняева Анна Андреевна

Группа: НПИмд-01-24

МОСКВА

2025

Теоретическое введение

Разложение на множители

pho-метод Полланда (или $\rho - 1$ метод Полларда) является одним из алгоритмов для факторизации целых чисел, который особенно эффективен для нахождения малых простых делителей. Он основан на свойствах чисел и использует последовательности, чтобы вычислить делители.

Основные этапы метода

1. Подготовка:
    - **Выбор числа n:** Начинаем с целого числа n, которое необходимо факторизовать;
    - **Выбор параметров:** Выбираем небольшое целое число a и границу B, которая будет использоваться для ограничения множителей.
2. Генерация последовательности: Создаем последовательность чисел по формуле: $x_{k+1} = (x_k^2 + a)$.
3. Вычисление НОД: На каждом шаге вычисляем наибольший общий делитель (НОД) между n и разностью двух членов последовательности.
4. Проверка результата: Если найденный НОД d больше 1 и меньше n, то это делитель числа n. Если $d = n$, то алгоритм не дал результата, и его можно повторить с другими параметрами. Если $d = 1$, то повторяем действия со второго шага.
5. Завершение: Процесс продолжается до тех пор, пока не будет найден делитель или не исчерпаются все возможные варианты.

Применение метода

Метод Полланда эффективен для нахождения малых простых делителей, особенно когда число имеет структуру, позволяющую выделить такие делители. Он также может быть использован в сочетании с другими методами факторизации для повышения общей эффективности.

Цель работы

Знакомство и реализация алгоритма разложения числа на множители.

Ход работы

1. Бинарный алгоритм Евклида

```
function evklidBin(a, b)
    if a == 0 || b == 0
        return 0
```

```
    elseif a == b
        return a
    elseif a < 0
        a *= -1
    elseif b < 0
        b *= -1
    end
    g = 1
    u = a; v = b
    while u > 0
        if u % 2 == 0 && v % 2 == 0
            g *= 2
            u = round(Int, u/2)
            v = round(Int, v/2)
        elseif u % 2 == 0
            u = round(Int, u/2)
        elseif v % 2 == 0
            v = round(Int, v/2)
        elseif u >= v
            u = u - v
        else
            v = v - u
        end
    end
    g *= v
    return g
end
```

Что происходит в функции: - НОД(0, b) = b, НОД(a, 0) = a - Если a и b четные: НОД(a, b) = 2 * НОД(a/2, b/2) - Если a четное, b нечетное: НОД(a, b) = НОД(a/2, b) - Если оба нечетные: НОД(a, b) = НОД(|a-b|, min(a, b))

Бинарный алгоритм Евклида в p-методе Полларда нужен для:

1. Эффективного вычисления НОД(a-b, n)

- На каждой итерации p-метода нужно находить НОД разности между двумя элементами последовательности и исходного числа n
- Если НОД > 1 и < n, значит найден нетривиальный делитель

2. Оптимизации производительности

- Бинарный алгоритм работает быстрее классического Евклида для больших чисел
- Использует битовые сдвиги вместо деления, что эффективнее

3. Обнаружения делителя

- В p-методе: если a ≡ b (mod p) для некоторого делителя p числа n, то НОД(a-b, n) будет кратен p
- Когда “быстрый” и “медленный” указатели встречаются в цикле по модулю p, их разность делится на p

2. ρ-метод Полларда

```
function metodPollarda(n, c, any_func::Function)
    if n % 2 == 0
        return 2, round(Int, n/2)
    end
    a = c; b = c
    i = 0
    p = 0
    while p == 0 && i < 100
        a = any_func(a)
        b = any_func(any_func(b))
        d = evklidBin(a-b, n)
        if d > 1
            return d, round(Int, n/d)
        end
        i += 1
    end
    return "Делитель не найден"
end
```

Что происходит в функции: - Ищет нетривиальные делители составного числа

- Использует алгоритм Флойда для обнаружения циклов
- Один указатель движется медленно, другой - быстро
- При обнаружении цикла вычисляет НОД разности и исходного числа
- Возвращает найденный делитель или сообщение об ошибке

3. Тестирование

```
n = 222
c = 1
metodPollarda(n, c, x -> (x^2 + 5) % n)
```

Результат тестирования представлен на рис.1

```
[4]: n = 222
     c = 1
     metodPollarda(n, c, x -> (x^2 + 5) % n)

[4]: (2, 111)
```

Рис. 1 Тестирование:

Вывод: В ходе данной работы мной были изучен и реализован p-метод Полларда для разложения числа на множетели. Написан программный код на языке Julia и протестирован.