Лабораторная работа №6. Разложение чисел на множители

Выполнила: Лебедева Ольга Андреевна

РУДН, Москва, Россия

2025

Цель работы

Изучить и реализовать на языке Julia[1] алгоритм разложения составных чисел на множители, в частности ρ-метод Полларда[2], а также посмотреть его эффективность на различных числах.

Задачи

1. Ознакомиться с теоретическими основами задачи факторизации чисел.
2. Изучить идею ρ-метода Полларда.
3. Реализовать данный метод на языке Julia.
4. Проверить работу алгоритма на примера чисел различного типа.
5. Проанализировать результаты и сделать выводы о его применимости.

Объект и предмет исследования

Объект исследования: алгоритм факторизации составных чисел. Предмет исследования: вероятностные методы разложения чисел на множители, основанные на свойствах модульной арифметики.

Условные обозначения и термины

Факторизация (разложение на множители) — представление составного числа (n) в виде произведения простых чисел.

НОД(a, b) — наибольший общий делитель чисел a и b.

Модульная арифметика — система вычислений, в которой операции производятся по остатку от деления на заданное число n.

ρ-метод Полларда — стохастический (вероятностный) метод поиска нетривиальных делителей числа, использующий повторяющуюся последовательность в модульной арифметике и вычисление НОД разности элементов.

Техническое оснащение и выбранные методы проведения работы

Программное обеспечение:

- Язык программирования Julia.
- Среда разработки JupyterLab / VS Code.

Методы:

- Использование функции $f(x) = (x^2 + c) \bmod n$.
- Генерация последовательностей и вычисление НОД с числом n.
- Анализ полученных итераций и контроль циклов.

Теоретическое введение

Задача разложения числа на множители является фундаментальной для теории чисел и криптографии.
От её сложности зависит безопасность криптосистем с открытым ключом (например, RSA), поскольку нахождение делителей большого числа, являющегося произведением двух простых, требует значительных вычислительных затрат.

ρ-метод Полларда (читается «ро-метод») — один из простейших и наиболее эффективных вероятностных алгоритмов факторизации. Его идея основана на поиске цикла в последовательности, построенной по функции $f(x) = (x^2 + c) \bmod n$. Если два элемента последовательности совпадают по модулю одного из делителей n, то разность между ними будет кратна этому делителю. Используя вычисление НОД, можно получить нетривиальный делитель числа n.

Метод назван ρ (ро) из-за формы траектории последовательности на графике — она напоминает греческую букву ρ. Главное преимущество алгоритма — высокая скорость нахождения делителя при малом объёме памяти.

Задание

1. Реализовать р-метод Полларда в среде Julia.
2. Протестировать программу на различных числах.
3. Оформить результаты в табличном виде, аналогично примеру из методички.

Выполним задание с помощью языка Julia:

using Printf using Random

```
function pollard(n::Int; c::Int=1)
    f(x) = (x^2 + 5) % n
    a = c
    b = c
    d = 1
    i = 1
   println(@sprintf("%-5s %-10s %-10s %-10s", "i", "a", "b", "H
b,n)"))
    println("-"^35)
    while d == 1
        a = f(a)
        b = f(f(b))
        d = gcd(abs(a - b), n)
      println(@sprintf("%-5d %-10d %-10d %-10d", i, a, b, d))
        i += 1
    end
```

```
    if d == n
     println("\nДелитель не найден. Попробуйте другое значение
        return nothing
    else
        println("\nНетривиальный делитель найден: $d")
        return d
    end
end

n = 1359331
c = 1
pollard(n; c=c)
```

ρ-метод Полларда

Проверим результат работы кода для числа из методички 1359331. Как видно, уже на нескольких итерациях алгоритм находит нетривиальный делитель: См. рис. 1

```
     i      a           b           НОД(a-b,n)
[32]: 1181


     ------------------------------------
     1      6           41          1
     2      41          123939      1
     3      1686        391594      1
     4      123939      438157      1
     5      435426      582738      1
     6      391594      1144026     1
     7      1090062     885749      1181

     Нетривиальный делитель найден: 1181
```

Рис. 1: ρ-метод Полларда

ρ-метод Полларда

Попробуем подставить другое число, 9973. Для него алгоритм Полларда не нашёл нетривиальных делителей, что подтверждает его простоту: См. рис. 2

```
n = 9973
c = 1
pollard(n; c=c)
```

```
161   8661     4688     1
162   5993     5184     1
163   3281     4688     1
164   4099     5184     1
165   7274     4688     1
166   4316     5184     1
167   8270     4688     1
168   8044     5184     1
169   1117     4688     1
170   1069     5184     1
171   5844     4688     1
172   4789     5184     1
173   6599     4688     1
174   4688     5184     1
175   6830     4688     1
176   5184     5184     9973

Делитель не найден. Попробуйте другое значение c.
```

Рис. 2: ρ-метод Полларда

ρ-метод Полларда

Метод Полларда использует идею повторяющихся остатков в модульной арифметике.
Когда последовательность (f(x)) начинает повторяться по одному из скрытых модулей (например, по p, если (n = pq)), два значения (a) и (b) становятся равны по модулю p, но различаются по модулю q.
В этот момент разность (|a - b|) делится на p, а вычисление НОД выявляет этот делитель.

Полученные результаты и заключение

В ходе лабораторной работы был изучен и реализован ρ-метод Полларда для разложения составных чисел на множители.
Реализация на Julia позволила пронаблюдать процесс поиска делителя по шагам и убедиться в эффективности алгоритма.

Метод Полларда продемонстрировал свою простоту и надёжность при решении задач факторизации. Он особенно полезен при анализе чисел средней длины и может служить основой для более сложных алгоритмов факторизации, применяемых в криптографии.

Библиографическая справка

[1] Julia

[2] ρ-метод Полларда