marp: true css: custom-theme.css title: "Лабораторная работа №6"на простоту" subtitle: "Дисциплина: Математические основы защиты информации и информационной безопасности" author: Хосе Фернандо Леон Атупанья, НФИмд-01-24, 1032249918 institute: Российский Университет Дружбы Народов, Москва, Россия date: 26 октября 2024

Лабораторная работа №6

Дисциплина: Математические основы защиты информации и информационной безопасности

Тема: Разложение чисел на множители

Студент: Леон Фернандо Хосе Фернандо

Цель работы

Ознакомиться с алгоритмом разложения чисел на множители. И написать код, соответствующий этому процессу (лабораторная работа 6).

Задание

1. Реализовать рассмотренные алгоритмы программно
2. Разложить на множители данное преподавателем число

2. Выполнение лабораторной работы

1. Алгоритм, реализующий p-метод Полларда

В этой отчете следующий код реализует p-метод Полларда для целочисленной факторизации. Этот алгоритм определяет нетривиальный множитель заданного целого числа n, используя псевдослучайную функцию f(x) со сжимающими свойствами. Ниже приведена подробная реализация в Julia с последующим объяснением.

1. Алгоритм p-метод Полларда (1/4)

```
using Random
using Printf

# Function to compute the GCD (Greatest Common Divisor)
function gcd(a, b)
    while b != 0
        (a, b) = (b, a % b)
    end
```

```
    return abs(a)
end
```

1. Алгоритм p-метод Полларда (2/4)

```
function pollards_p_method(n::Int, c::Int, f::Function)
    # Step 1: Initialize a and b
    a = c
    b = c

    while true
        # Step 2: Update a and b using the function f
        a = f(a) % n
        b = f(f(b) % n) % n

        # Step 3: Compute d = GCD(a - b, n)
        d = gcd(abs(a - b), n)
```

1. Алгоритм p-метод Полларда (3/4)

```
        # Step 4: Check conditions for termination
        if d > 1 && d < n
            return d # Non-trivial divisor found
        elseif d == n
            return "No divisor found"
        end
        # If d == 1, continue the loop
    end
end
```

1. Алгоритм p-метод Полларда (4/4)

```
    # Example parameters
n = 1359331
c = 1
f(x) = (x^2 + 5)

# Run the algorithm
result = pollards_p_method(n, c, f)
```

```
# Print the result
if typeof(result) == Int
    println("A non-trivial divisor of $n is $result.")
else
    println(result)
end
```

Вывод

В этом упражнении p-метод Полларда был реализован в Julia для разложения целых чисел на множители. Алгоритм успешно продемонстрировал свою способность находить нетривиальные делители составных чисел, используя псевдослучайные итеративные обновления и свойства наибольшего общего делителя. Используя пример с n=1359331, алгоритм определил 1181 как нетривиальный фактор, подтверждающий его эффективность.