

Отчёт по лабораторной работе №6

Аветисян Давид Артурович

23 ноября 2024

РУДН, Москва, Россия

Реализовать алгоритм, реализующий р-метод Полларда.

- 1-й блок. Импорт необходимых модулей
- 2-й блок. Реализация самого алгоритма: Задаются начальные значения x и y и НОД d . С помощью цикла вычисляется медленный и быстрый шаги x и y . Затем они сравниваются, чтобы найти НОД. В конце проверяется результат выполнения алгоритма.
- 3-й блок. Задаётся значение n , для которого нужно найти делитель и вызывается функция, описанная выше.
- 4-й блок. Вывод ответа в зависимости от результат выполнения алгоритма.

Алгоритм, реализующий р-метод Полларда, на языке Julia

```
[1]: using Printf

[2]: function pollard_p(n, f = x -> (x^2 + 1) % n, x0 = 2)
    x = x0
    y = x0
    d = 1

    while d == 1
        x = f(x)
        y = f(f(y))

        d = gcd(abs(x - y), n)
    end

    if d == n
        return :failure
    else
        return d
    end
end

[2]: pollard_p (generic function with 3 methods)

[3]: n = 8051
divisor = pollard_p(n)

if divisor != :failure
    @printf("Найден делитель числа %d: %d\n", n, divisor)
else
    @printf("Метод не нашёл делителя числа %d\n", n)
end

Найден делитель числа 8051: 97

[4]: n = 1359331
divisor = pollard_p(n)

if divisor != :failure
    @printf("Найден делитель числа %d: %d\n", n, divisor)
else
    @printf("Метод не нашёл делителя числа %d\n", n)
end

Найден делитель числа 1359331: 1153
```

Рис. 1: Алгоритм, реализующий р-метод Полларда, на языке Julia

Я реализовал алгоритм, реализующий p -метод Полларда.