Лабораторная работа №6

Разложение чисел на множители

Тазаева Анастасия Анатольевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Ознакомиться с pho-методом Полларда. Реализовать его.

2 Задание

1. Реализовать на языке программирования Julia pho-метод Полларда.
2. Разложить на множители число 1359331.

3 Теоретическое введение [2]

3.1 Разложение на множители

pho-метод Полларда (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.

3.1.1 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.

3.1.2 Применение метода

Метод Полларда 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.

4 Выполнение лабораторной работы

4.1 pho-метод Полларда

Написан программный код на языке `Julia` [1], реализующий pho-метод Полларда:

```
function pollard(n, c, f::Function)
    # step 1
    a = c
    b = c
    # step 2-4
    println(a, "\t", b, "\t")
    while true
        # step 2
        a = f(a) % n
        b = f(b) % n
        b = f(b) % n

        # step 3
        d = binary_gcd(abs(a-b),n)

        println(a, "\t", b, "\t", d)
        # step 4
```

```
        if 1 < d < n
            return d, round(Int, n/d)
        elseif d == n
            return "Delitel ne naiden"
        end
    end
end
```

Получен следующий результат выполнения программного кода (рис. 4.1).

```
n = 1359331
c = 1
pollard(n, c, x -> (x^2 + 5) % n)
```

```
1       1
6       41      1.0
41      123939  1.0
1686    391594  1.0
123939  438157  1.0
435426  582738  1.0
391594  1144026 1.0
1090062 885749  1181.0
```

```
[20]:

(1181.0, 1151)
```

Рисунок 4.1: pho-метод Полларда. Пример

5 Выводы

В ходе лабораторной работы был изучен pho-метод Полларда.

Список литературы

[1] *Julia 1.10 Documentation*. Англ. 2024. URL: https://docs.julialang.org/en/v1/.

[2] *Математика криптографии и теория шифрования*. URL: https://intuit.ru/studies/courses/552/408/info.