Лабораторная работа №6

Разложение чисел на множители

Тазаева Анастасия Анатольевна

2025-11-08

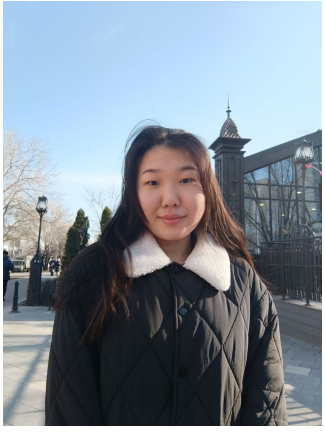
Содержание I

Раздел 1

1. Информация

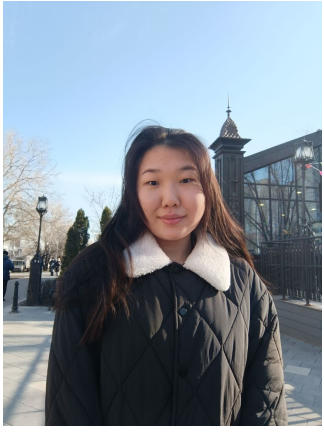
1.1 Докладчик

- Тазаева Анастасия Анатольевна



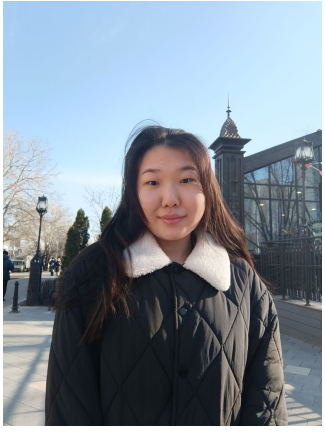
1.1 Докладчик

- Тазаева Анастасия Анатольевна
- студент группы НФИмд-02-25



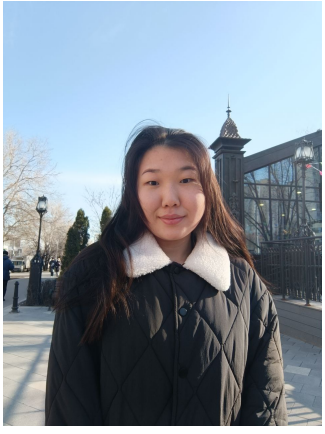
1.1 Докладчик

- Тазаева Анастасия Анатольевна
- студент группы НФИмд-02-25
- Российский университет дружбы народов им. П. Лумумбы



1.1 Докладчик

- Тазаева Анастасия Анатольевна
- студент группы НФИмд-02-25
- Российский университет дружбы народов им. П. Лумумбы
- 1032259385@pfur.ru



Раздел 2

2. Введение

2.1 Цель работы

Ознакомиться с pho-методом Полларда. Реализовать его.

2.2 Задачи

1. Реализовать на языке программирования Julia pho-метод Полларда.

2.2 Задачи

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

Раздел 3

3. Теоретические сведения

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.

3.2 Основные этапы метода

1. Подготовка:

3.2 Основные этапы метода

1. Подготовка:
   - **Выбор числа n:** Начинаем с целого числа n, которое необходимо факторизовать;

3.2 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.

3.2 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: $x_{k+1} = (x_k^2 + a)$.

3.2 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.

3.2 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.

3.2 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.

Раздел 4

4. Программный код

4.1 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
```

4.2 pho-метод Полларда

```
        # 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
```

4.3 pho-метод Полларда. Результат работы программного кода

```
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)
```

Раздел 5

5. Заключение

5.1 Вывод

В ходе лабораторной работы был изучен pho-метод Полларда.