Лабораторная работа №5

Вероятностные алгоритмы проверки чисел на простоту

Топонен Н. А.

4 ноября 2023

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Топонен Никита Андреевич
- студент Российского университет дружбы народов
- 1132236933@rudn.ru
- https://github.com/natoponen

Вводная часть

Цель работы

- Изучить вероятностные алгоритмы проверки чисел на простоту.

Задание

Реализовать алгоритмы:

1. Алгоритм, реализующий тест Ферма;
2. Алгоритм вычисления символа Якоби;
3. Алгоритм, реализующий тест Соловэя-Штрассена;
4. Алгоритм, реализующий тест Миллера-Рабина.

Теоретическое введение

Тест простоты

- Тестом простоты (или проверкой простоты) называется алгоритм, который, приняв на входе число N, позволяет либо не подтвердить предположение о том, является ли это число составным, либо точно утверждать его простоту.

1. Тест Ферма.
2. Тест Миллера — Рабина.
3. Тест Соловэя — Штрассена.
4. Тест Бейли — Померанца — Селфриджа — Уогстаффа.
5. Квадратичный тест Фробениуса.

Выполнение лабораторной работы

Вспомогательные функции

```java
private int getRandomInt(int min, int max) {
    Random random = new Random();
    return random.nextInt( bound: max - min) + min;
}

5 usages
public int modPow(int a, int b, int c) {
    int res = 1;
    for (int i = 0; i < b; i++) {
        res *= a;
        res %= c;
    }
    return res % c;
}
```

Рис. 1: Вспомогательные функции

Тест Ферма

```
private String fermat(int n) {
    if (n % 2 == 0 && n < 5) {
        throw new RuntimeException("n must uneven and be greater or equal than 5");
    }

    int a = getRandomInt(2, n - 2);
    int r = modPow(a,  b: n - 1, n);

    if (r == 1) {
        return String.format("%s is probably prime", n);
    } else {
        return String.format("%s is probably composite", n);
    }
}
```

Рис. 2: Тест Ферма

Вычисления символа Якоби

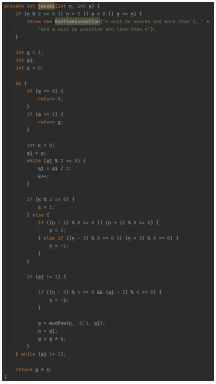

Рис. 3: Вычисления символа Якоби

Тест Соловэя — Штрассена

```java
private String solovejShtrassen(int n) {
    if (n % 2 == 0 || n < 5) {
        throw new RuntimeException("n must be uneven and more than 4");
    }

    int a = getRandomInt(2, n - 2);
    int r = modPow(a,  b: (n-1)/2, n);

    if (r != 1 && r != (n - 1)) {
        return String.format("%s is composite", n);
    }

    int s = jacobi(n, a);
    if ((r - s) % n == 0) {
        return String.format("%s is composite", n);
    } else {
        return String.format("%s is probably prime", n);
    }
}
```

Рис. 4: Тест Соловэя — Штрассена

Тест Миллера — Рабина

```java
private String millerRabin(int n) {
    if (n % 2 == 0 || n < 5) {
        throw new RuntimeException("n must be uneven and more than 4");
    }

    int s = 0;
    int r = 0;
    int nEven = n - 1;

    while (nEven % 2 == 0) {
        nEven = nEven / 2;
        s++;
    }
    r = nEven;

    int a = getRandomInt(2, n-2);
    int y = modPow(a, r, n);

    if (y != 1 && y != (n - 1)) {
        for (int i = 1; i <= (s - 1) && y != (n - 1) && y != 1; i++) {
            y = modPow(y,  b: 2, n);
        }
        if (y == 1 || y != (n - 1)) {
            return String.format("%s is composite", n);
        }
    }

    return String.format("%s is probably prime", n);
}
```

Рис. 5: Тест Миллера — Рабина

Выводы

- Изучил и реализовал вероятностные алгоритмы проверки чисел на простоту.