Лабораторная работа №5

Математические основы защиты информации и информационной безопасности

Минов К.В, НПМмд-02-2319

октября 2023

Российский университет дружбы народов

Москва, Россия

Цель лабораторной работы

1) Реализовать на языке программирования вероятностные алгоритмы проверки чисел на простоту

Пусть a - целое число. Числа ± 1, $\pm a$ называются тривиальными делителями числа a.

Целое число $p \in Z/\{0\}$ называется простым, если оно не является делителем единицы ине имеет других делителей, кроме тривиальных. В противном случае число $p \in Z/\{-1, 0, 1\}$ называется составным.

Детерминированный алгоритм всегда действует по одной и той же схеме и гарантированно решает поставленную задачу (или не дает никакого ответа).

Вероятностный алгоритм использует генератор случайных чисел и дает не гарантированно точный ответ. Вероятностные алгоритмы в общем случае не менее эффективны, чем детерминированные (если используемый генератор случайных чисел всегда дает набор одних и тех же чисел, зависящих от входных данных, то вероятностный алгоритм становится детерминированным).

Для проверки на простоту числа n вероятностным алгоритмом выбирают случайное число a $(1 < a < n)$ и проверяют условия алгоритма. Если число n не проходит тест по основанию a, то алгоритм выдает результат “Число n составное”, и число n действительно является составным.

Если же n проходит тест по основанию a, ничего нельзя сказать о том, действительно ли число n является простым. Последовательно проведя ряд проверок таким тестом для разных a и получив для каждого из них ответ “Число n, вероятно, простое”, можно утверждать, что число n является простым с вероятностью, близкой к 1.

Ход выполнения лабораторной работы

- Реализуем тест Ферма

```java
import java.util.Random;

public class FermatTest {

    public static String fermat(int n) {
        Random random = new Random();
        int a = random.nextInt( bound: (n - 2) + 1) + 2; // Генерация случайного числа в диапазоне [2, n-2]
        int r = modPow(a,  b: n - 1, n); // Вычисление a^(n-1) mod n

        if (r == 1) {
            return "число " + n + " вероятно простое";
        } else {
            return "число " + n + " составное";
        }
    }

    // Вспомогательная функция для вычисления a^b mod m
    private static int modPow(int a, int b, int m) {
        int result = 1;
        a = a % m;
        while (b > 0) {
            if (b % 2 == 1) {
                result = (result * a) % m;
            }
            b /= 2;
            a = (a * a) % m;
        }
        return result;
    }

    public static void main(String[] args) {
        int numberToTest = 15; // Замените на нужное число для теста
        System.out.println(fermat(numberToTest));
    }
}
```

Figure 1: Рис.1: Тест Ферма

Ход выполнения лабораторной работы

- Реализуем вычисление символа Якоби

Figure 2: Рис.2: Вычисление символа Якоби

Ход выполнения лабораторной работы

- Реализуем тест Соловэя-Штрассена

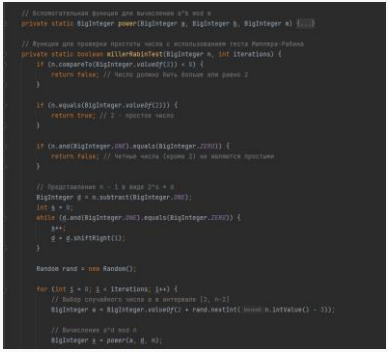

Figure 3: Рис.3: Тест Соловэя-Штрассена

Ход выполнения лабораторной работы

- Реализуем тест Миллера-Рабина

Figure 4: Рис.4: Тест Миллера-Рабина

Вывод

- В ходе выполнения данной лабораторной работы были реализованы вероятностные алгоритмы проверки чисел на простоту