Лабораторная работа №5

Вероятностные алгоритмы проверки чисел на простоту

Цель работы

Реализация и исследование вероятностных алгоритмов проверки чисел на простоту:

1. Тест Ферма
2. Тест Соловэя-Штрассена
3. Тест Миллера-Рабина

Теоретическая часть

Основные определения

- **Простое число** - натуральное число, имеющее ровно два делителя: 1 и само число
- **Составное число** - натуральное число, имеющее более двух делителей
- **Вероятностный алгоритм** - алгоритм, использующий случайность и дающий правильный ответ с высокой вероятностью

Алгоритмы проверки простоты

Тест Ферма

Основан на малой теореме Ферма: если p - простое число, то для любого a, взаимно простого с p: a^(p-1) ≡ 1 (mod p)

Тест Соловэя-Штрассена

Использует критерий Эйлера и символ Якоби. Для простого p и любого a: a^((p-1)/2) ≡ (a/p) (mod p)

Тест Миллера-Рабина

Более мощный тест, основанный на разложении n-1 = 2^s * r и проверке последовательных возведений в квадрат.

Практическая реализация

Структура программы

- `primality_tests.py` - основные алгоритмы проверки простоты
- `test_comprehensive.py` - расширенное тестирование и анализ

Результаты тестирования

Известные простые числа

Все три алгоритма корректно идентифицируют малые простые числа (5, 7, 11, 13, ...)

Известные составные числа

Все алгоритмы корректно идентифицируют малые составные числа (9, 15, 21, 25, ...)

Числа Кармайкла

- Тест Ферма: часто ошибается, принимая числа Кармайкла за простые
- Тест Соловэя-Штрассена: более надежен, но также может ошибаться
- Тест Миллера-Рабина: наиболее надежен, правильно идентифицирует числа Кармайкла как составные

Анализ вероятности ошибки

Вероятность ошибки после t итераций:

- Тест Ферма: ≤ 1/2^t (для большинства чисел)
- Тест Соловэя-Штрассена: ≤ 1/2^t
- Тест Миллера-Рабина: ≤ 1/4^t

Выводы

1. Тест Миллера-Рабина является наиболее надежным из трех реализованных алгоритмов
2. Числа Кармайкла представляют наибольшую сложность для вероятностных тестов простоты
3. Увеличение количества итераций значительно снижает вероятность ошибки
4. Для криптографических применений рекомендуется использовать тест Миллера-Рабина с 20+ итерациями

Использование программы

```
# Основное тестирование
python primality_tests.py

# Расширенное тестирование
python test_comprehensive.py
```

```
<style type="text/css">@media print {
    *, :after, :before {background: 0 0 !important;color: #000 !important;box-shadow: none !important;text-shadow: none !im
    a, a:visited {text-decoration: underline}
    a[href]:after {content: " (" attr(href) ")"}
    abbr[title]:after {content: " (" attr(title) ")"}
    a[href^="#"]:after, a[href^="javascript:"]:after {content: ""}
    blockquote, pre {border: 1px solid #999;page-break-inside: avoid}
    thead {display: table-header-group}
    img, tr {page-break-inside: avoid}
    img {max-width: 100% !important}
    h2, h3, p {orphans: 3;widows: 3}
    h2, h3 {page-break-after: avoid}
}
html {font-size: 12px}
@media screen and (min-width: 32rem) and (max-width: 48rem) {
    html {font-size: 15px}
}
@media screen and (min-width: 48rem) {
    html {font-size: 16px}
}
body {line-height: 1.85}
.air-p, p {font-size: 1rem;margin-bottom: 1.3rem}
.air-h1, .air-h2, .air-h3, .air-h4, h1, h2, h3, h4 {margin: 1.414rem 0 .5rem;font-weight: inherit;line-height: 1.42}
.air-h1, h1 {margin-top: 0;font-size: 3.998rem}
.air-h2, h2 {font-size: 2.827rem}
.air-h3, h3 {font-size: 1.999rem}
.air-h4, h4 {font-size: 1.414rem}
.air-h5, h5 {font-size: 1.121rem}
.air-h6, h6 {font-size: .88rem}
.air-small, small {font-size: .707em}
canvas, iframe, img, select, svg, textarea, video {max-width: 100%}
body {color: #444;font-family: 'Open Sans', Helvetica, sans-serif;font-weight: 300;margin: 0;text-align: center}
img {border-radius: 50%;height: 200px;margin: 0 auto;width: 200px}
a, a:visited {color: #3498db}
a:active, a:focus, a:hover {color: #2980b9}
pre {background-color: #fafafa;padding: 1rem;text-align: left}
blockquote {margin: 0;border-left: 5px solid #7a7a7a;font-style: italic;padding: 1.33em;text-align: left}
li, ol, ul {text-align: left}
p {color: #777}</style>
```