Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

**Лабораторная работа №5**

студента 5 курса 531 группы

специальности 10.05.01 «Компьютерная безопасность»

факультета компьютерных наук и информационных технологий

Енца Михаила Владимировича

|  |  |  |
| --- | --- | --- |
| Преподаватель  профессор | \_\_\_\_\_\_\_\_\_\_ | В. А.  Молчанов |
|  | подпись, дата |  |
| Заведующий кафедрой  д.ф.-м.н., доцент | \_\_\_\_\_\_\_\_\_\_ | М. Б. Абросимов |
|  | подпись, дата |  |

Саратов 2018

**1. Постановка задачи.**

Изучение основных методов факторизации целых чисел и их программная реализация.

**2. Теоретические сведения по рассмотренным темам с их обоснованием.**

*Определение.* Задача разложения составного числа на множители формулируется так: для данного положительного целого числа найти его каноническое разложение , где – попарно различные простые числа, .

-Метод Полларда

Пусть – нечетное составное число, и –случайное отображение, обладающее сжимающими свойствами, например . Основная идея метода состоит в следующем. Выбираем случайный элемент и строим последовательность определяемую рекуррентным соотношением , где , до тех пор, пока не найдем такие числа , что и . Поскольку множество конечно, такие индексы существуют (последовательность «зацикливается»). Последовательность будет состоять из «хвоста» длины и цикла той же длины.

Если – простой делитель числа и , то разность делится на и на . Число нам нужно найти, поэтому все вычисления в алгоритме будем проводить по модулю и на каждом шаге вычислять . Нетривиальный наибольший общий делитель (когда числа и принадлежат одному классу вычетов по модулю , но разным классам вычетов по модулю ) как раз и будет искомым делителем числа . Случай имеет место с пренебрежимо малой вероятностью.

(p-1)-Метод Полларда.

Пусть – нечетное составное число и – его нетривиальный делитель. (p-1)-Метод особенно эффективен при разложении таких чисел , для которых число сильно составное.

Определение.

Пусть – множество различных простых чисел. Назовем множество базой разложения. Целое число назовем -гладким, если все его протсые делители являются элементами множества .

Пусть и пусть каноническое разложение числа имеет вид . Найдем максимальные показатели , для которых . Прологарифмируем обе части этого неравенства: , откуда , тогда для некоторого целого числа .

Согласно малой теореме Ферма, выполняется сравнение для любого целого , взаимно простого с . Возводя обе части этого сравнения в степень , получаем

Обозначим Если то число должно делиться на , поскольку разность делится на и число делится на .

Метод непрерывных дробей

Теорема. (Ферма, о разложении).

Для любого положительного нечетного числа существует взаимно однозначное соответствие между множеством делителей числа n, не меньших, чем , и множеством пар таких неотрицательных целых чисел, что .

Если , где числа и близки друг к другу, то число мало, а значит, немного больше, чем . В этом случае можно найти и , последовательно перебирая числа до тех пор, пока не найдется такое , что разность является полным квадратом, то есть равна .

Чем больше разность между числами и , тем более трудоемким становится метод Ферма. В этом случае можно воспользоваться обобщенным методом Ферма: для небольшого целого числа последовательно вычислять пока не получится такое число s, что разность является полным квадратом, то есть равна . Отсюда , и значит, числа и n имеют нетривиальный общий делитель.

Поскольку подобрать такое число не всегда легко, на практике для разложения числа достаточно найти такие целые числа , что , то есть . Если то число n делит произведение двух чисел и , но не делит ни один из сомножителей. Значит, один делитель числа делит разность , а другой делитель делит сумму .

В методе непрерывных дробей используется модернизация алгоритма Диксона. В качетве чисел выбираются числители подходящих дробей к обыкновенной дроби, выражающей число .

В алгоритме Диксона в качестве можно брать числители подходящих дробей к . Кроме того, из базы можно исключить те простые числа , по модулю которых является квадратичным невычетом.

**3. Результаты работы.**

**Псевдокоды рассмотренных алгоритмов**

Псевдокод -Метод Полларда.

*Вход.* Число , начальное значение , функция , обладающая сжимающими свойствами.

*Выход.* Нетривиальный делитель числа .

1. Положить .

2. Вычислить

3. Найти

4. Если , то положить и результат: . При результат: «Делитель не найден»; при вернуться на шаг 2.

Псевдокод (p-1)-Метод Полларда.

*Вход*. Составное число .

*Выход*. Нетривиальный делитель числа .

1. Выбрать базу разложения .

2. Выбрать случайное целое , и вычислить . При положить и результат: .

3. Для выполнить следующие действия.

3.1. Вычислить .

3.2. Положить

4. Вычислить

5. при или результат: «Делитель не найден». В противном случае положить и результат: .

Псевдокод Метод непрерывных дробей.

*Вход.* Составное число .

*Выход.* Нетривиальный делитель p числа n.

1. Построить базу разложения где и -- попарно различные простые числа.

1.1. Исключить из те простые числа , по модулю которых является квадратичным невычетом.

2. Найти целых чисел , для каждого из которых абсолютно наименьший вычет является -гладким:

Где , и каждому числу сопоставить вектор показателей .

3. Найти (например методом гауссова исключения) такое непустое множество что , где .

4. Положить . Тогда .

5. Если то положить и результат: В противном случае вернуться на шаг 3 и поменять множество .

**Коды программ, реализующей рассмотренные алгоритмы**

Программа реализована на языке Python (версия интерпретатора 3.6).

**Оценки сложности рассмотренных алгоритмов**

Сложность алгоритма -Метод Полларда равна .

Сложность алгоритма -Метод Полларда равна .

Временная сложность алгоритма Диксона, использующего непрерывные дроби, равна .

**Результаты тестирования программ**

В ходе работы были рассмотрены следующие методы разложения целых числе на множители: метод непрерывных дробей, -метод Полларда, -метод Полларда. Так как в общем случае проверка числа на простоту является менее трудоемкой, чем любой из алгоритмов разложения числа на множители, то имеет смысл каждое число перед разложением на множители проверить на простоту.

Пример работы алгоритмов (анализ корректности)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Делитель | | |
| Число | -Метод Полларда | -Метод Полларда | Метод непрерывных дробей |
|  |  |  |  |
|  | Число прошло тест на простоту! | | |
|  |  |  |  |

Примеры работы алгоритмов (анализ эффективности)

Для сравнительно анадиза алгоритмов Полларда целесобразно воспользоваться тестом, аналогичным одному из тестов модификаций алгоритма Евклида.

Для фиксированного числа оцениваем время работы алгоритмов (на 1000 запусков). При такой оценки многое зависит как от вида числа, так и от свойств выбранного отображения. В данном случае оно представляло собой функцию Эту функцию достаточно легко вычислить. Очевидно, такая реализация p-метода Полларда будет выигрывать по сравнению с реализацией -метода хотя бы потому, что последняя требует (помимо операций умножения и деления) вычисления экспоненты и трехкратного вычисления логарифма на каждой итерации. Действительно, -метод Полларда превосходит -метод более чем в 10 раз для числа любого вида.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | -метод Полларда | | -Метод Полларда | |
| Число | Делитель | Время | Делитель | Время |
| 11021 |  |  |  |  |
| 10000 |  |  |  |  |

Несмотря на эффективность, -метод Полларда имеет существенный недостаток: если случайное отображение фиксированно, то для некоторых чисел он не работает. В этом случае требуется поменять отображение или воспользоваться иным методом разложения на множители.