Лабораторная работа № 1
ФАКТОРИЗАЦИЯ СОСТАВНОГО ЧИСЛА

ЦЕЛЬ РАБОТЫ

Освоить простые алгоритмы факторизации составного числа.

УКАЗАНИЕ К РАБОТЕ

Ознакомиться с приведенными ниже методическими указаниями, а также с литературой [2, 13–17].

Для криптографического вскрытия алгоритма шифрования RSA достаточно разложить часть открытого ключа на простые множители, поэтому задача факторизации составного числа приобрела большое практическое значение. Данная задача является обратной к задаче определения простоты конкретного числа, но значительно сложнее.

Далее представлены наиболее простые методы факторизации составного числа.

МЕТОД ФЕРМА

Данный метод основан на поиске таких чисел x и y, что $x^2 \equiv \equiv y^2 \pmod{n}$, где n надо разложить на множители.

Теорема 1 (Эйлера о представлении числа в виде разности квадратов)

Если $n > 1$ нечетно, то существует взаимно однозначное соответствие между разложениями на множители $n = a \cdot b$ и представлениями в виде разности квадратов $n = x^2 - y^2$, $x > y > 0$. Здесь $x = \frac{a+b}{2}$, $y = \frac{a-b}{2}$, $a = x + y$, $b = x - y$.

Метод Ферма заключается в том, что при малых значениях параметра y в представлении $n = x^2 - y^2$ можно найти пару (x, y), перебирая

в качестве кандидатов на значение x числа $\left\lfloor \sqrt{n}+1 \right\rfloor$, $\left\lfloor \sqrt{n}+2 \right\rfloor$, … и проверяя для каждого из них равенства $\left(\left\lfloor \sqrt{n}+j \right\rfloor \right)^2 - n = y^2$.

Алгоритм факторизации методом Ферма

Вход: n – нечетное число, $p_1, \ldots, p_k$ – небольшие простые числа.

1. Проверить, делят ли нацело p_k, $i = \overline{1,k}$ число n. Если да, то делитель найден (останов алгоритма).

2. Для каждого $x \in \left[\left\lceil \sqrt{n} \right\rceil ; \frac{n}{2}+1 \right]$ вычислить величину $t = x^2 - n$. Если были проверены все x из этого диапазона и ни один делитель не был найден, то число n – простое.

3. Проверить, является ли t полным квадратом. Если $t = y^2$, то n – составное и делитель найден ($a = x + y$, $b = x - y$, останов алгоритма); если t не является полным квадратом, то перейти к следующему x на шаге 2.

Другой вариант метода Ферма предложил Кнут (без операций деления).

Пусть $n = a \cdot b$, где $a \le b$.

Допустим, что n – нечетно, тогда a и b тоже нечетны. Поэтому можно положить $x = \frac{a+b}{2}$, $y = \frac{a-b}{2}$, $n = x^2 - y^2$, $0 \le y - x \le n$. Метод Ферма заключается в поиске x и y, удовлетворяющих этим соотношениям. Тогда $n = (x - y)(x + y)$.

Следующий алгоритм показывает, как, не выполняя операций деления, можно реализовать метод Ферма.

1. $x' = 2 \cdot \left\lfloor \sqrt{n} \right\rfloor + 1$, $y' = 1$, $r = \left\lfloor \sqrt{n} \right\rfloor^2 - n$ (во время работы алгоритма $x = (x' - 1)/2$, $y = (y' - 1)/2$, $r = x^2 + y^2 - n^2$).

2. Если $r \le 0$, то перейти на шаг 4.

3. $r = r - y'$, $y' = y' + 2$ и вернуться на шаг 2.

4. Если $r = 0$, то останов алгоритма: имеем $n = a \cdot b = \frac{x' - y'}{2} \cdot \frac{x' + y' - 2}{2}$ и значение $a = \frac{x' - y'}{2}$ есть наибольший делитель числа n, не превосходящий $\sqrt{n}$.

5. $r = r + x'$, $x' = x' + 2$ и вернуться на шаг 3.

Число шагов, выполняемых этим алгоритмом для нахождения a и b, пропорционально $\sqrt{n}$.

Метод Ферма эффективен, только если делители числа n близки друг к другу. Данный метод является, по сути, переборным, следовательно, неэффективным.

(p–1)-ФАКТОРИЗАЦИЯ ПОЛЛАРДА

Предположим, что n – нечетное составное число, не имеющее небольших простых делителей. Обозначим через p наименьший простой делитель числа n. Наша задача заключается в его нахождении.

Предположим, что число $p - 1$ разлагается в произведение небольших простых делителей. Выберем число k, которое является параметром метода. Для успешной работы алгоритма нужно, чтобы выполнялось условие: $p - 1$ делит $M(k)$, где $M(k) = \text{НОК}(1, 2, \ldots, k)$ (вместо $M(k)$ можно использовать, например, $k!$).

В силу малой теоремы Ферма выполняется сравнение $2^{M(k)} \equiv \equiv 1 \pmod{p}$. Если при этом $2^{M(k)} \equiv 1 \pmod{n}$, то p делит НОД $(2^{M(k)} - 1, n)$, где $p > 1$, НОД $(2^{M(k)} - 1, n) < n$.

Таким образом, НОД $(2^{M(k)} - 1, n)$ является делителем числа n, кратным p.

Так как число k неизвестно, то оно ищется в алгоритме перебором.

Алгоритм метода (p – 1)-факторизации Полларда

Пусть k – целое число, например, $k < 10^6$ и c – небольшое целое, для которого выполняется условие НОД $(c, n) = 1$, например, $c = 2$.

1. Для каждого $i = \overline{1, k}$ вычисляется $m_i = c^{M(i)} \bmod n$ и проверяется тест шага 2.

2. Вычисляется $d = \text{НОД}(m_i - 1, n)$. Если $1 < d < n$, то d – нетривиальный делитель числа n. В противном случае полагаем $i = i + 1$.

Оценка сложности данного метода в худшем случае составляет $O(n^{1/2} \cdot \log^{\text{const}} n)$ арифметических операций. Однако в некоторых случаях алгоритм может быстро выдать делитель числа n.

На практике ($p - 1$)-метод Полларда обычно используют до применения более сильных алгоритмов факторизации для того, чтобы отделить небольшие простые делители числа n.

МЕТОД ρ-ПОЛЛАРДА

Алгоритм

1. Случайным образом выбирается x_1 из множества $\{0, 1, \dots, n-1\}$. $y = x_1$, $k = 2$, $i = 1$.

2. $i = i + 1$. Вычисляется следующий элемент последовательности $x_i = f(x_{i-1}) \bmod n$, где $f(x) = x^2 + 1$.

3. Вычисляется $d = \text{НОД}(y - x_i, n)$. Если $1 < d < n$, то d является делителем n (останов алгоритма), иначе выполняется переход на шаг 4.

4. Если $i < k$, то осуществляется переход на шаг 2.

5. Если $i = k$, то $y = x_i$, $k = 2 \cdot k$ и выполняется переход на шаг 2.

Возможно, что цикл значений по модулю n окажется больше, чем $\sqrt{n}$. Метод имеет эвристическую оценку сложности $O(n^{1/4})$ арифметических операций. Он очень популярен и обычно используется для отделения небольших простых делителей факторизуемого числа n.

Основная идея данного метода очень проста. Если период последовательности $x_i \bmod n$ может быть порядка n, то период последовательности $x_i \bmod p$ для простого делителя p числа n не превосходит p. Это значит, что y и x_i могут быть различными по модулю n, но совпадать по модулю p.

Существует такая константа c, что для любого $\lambda > 0$ вероятность не найти нетривиальный делитель n за $c\sqrt{\lambda} \cdot n^{1/4} \log^3 n$ битовых операций будет меньше, чем $e^{-\lambda}$.

МЕТОД ЛЕМАНА

Алгоритм

Пусть n нечетно и $n > 8$.

1. Для $a = 2, 3, \dots, [n^{1/3}]$ проверить, что a делит n. Если на этом шаге найдется делитель числа n, то алгоритм заканчивает свою работу, иначе выполняется переход к шагу 2.

2. Для всех $k = 1, 2, \dots, [n^{1/3}]$ и всех $d = 0, 1, \dots, \left[\frac{n^{1/6}}{4\sqrt{k}}\right] + 1$ проверить, является ли число $\left(\left[\sqrt{4kn}\right] + d\right)^2 - 4kn$ квадратом натурального числа.

3. Если является, то для $A = \left[\sqrt{4kn}\right] + d$ и $B = \sqrt{A^2 - 4kn}$ выполнено сравнение $A^2 \equiv B^2 \pmod{n}$ (или $(A - B)(A - B) \equiv 0 \pmod{n}$). В этом случае вычисляется $d^* = \text{НОД}(A - B, n)$.

4. Если $1 < d^* < n$, то d^* и (n / d^*) – делители числа n. Алгоритм останавливается.

Если данный алгоритм не нашел разложение n на два множителя, то n – простое число.

Данный алгоритм раскладывает n на множители за O ($n^{1/3}$) арифметических операций.

МЕТОД ЛЕНСТРЫ

Теорема 2

Пусть r, s, n – натуральные числа, удовлетворяющие условиям $1 \le r < s < n$, $n^{1/3} < s$, НОД $(r, s) = 1$.

Тогда существует не более 11 делителей r_i числа n таких, что $r_i \equiv r \pmod{s}$. Имеется алгоритм, который находит все эти r_i за O (log n) арифметических операций.

Алгоритм метода факторизации Ленстры

На входе заданы числа r, s, $n \in N$, удовлетворяющие условиям теоремы.

1. С помощью расширенного алгоритма Евклида найти $r^* \in N$: $r^* r \equiv 1 \pmod{s}$. Затем найти r' такое, что $r' \equiv r^* \cdot n \pmod{s}$, $0 \le r' < s$.

2. Для очередного значения $i = 0, 1, \ldots$ найти числа a_i, b_i, c_i по следующим правилам:

$$a_0 = s,\ b_0 = 0,\ c_0 = 0,$$

$$a_1 \equiv r' r^* \pmod{s},\quad 0 < a_1 \le s,\ b_1 = 1,\ c_1 \equiv \frac{n - rr'}{s} r^* \pmod{s},$$

$$\forall i \ge 2:$$

$$a_i = a_{i-2} - q_i a_{i-1},\ b_i = b_{i-2} - q_i b_{i-1},\ c_i \equiv c_{i-2} - q_i c_{i-1} \pmod{s},$$

где числа q_i однозначно определяются условиями:

$$0 \le q_i < a_{i-1},\ i - \textit{четное},$$
$$0 < q_i \le a_{i-1},\quad i - \textit{нечетное}.$$

3. Для очередного набора a_i, b_i, c_i найти все целые числа c, удовлетворяющие условиям:

$$c \equiv c_i (\mathrm{mod}\, s),$$
$$|c| < s, \quad i - \textit{четное},$$
$$2a_i b_i \le c \le \frac{n}{s^2} + a_i b_i, \quad i - \textit{нечетное}.$$

Таких c будет не более двух.

4. Для каждого c из шага 3 решить в целых числах систему уравнений

$$\begin{cases} xa_i + yb_i = c, \\ (xs + r)(ys + r') = n. \end{cases}$$

Если x и y окажутся неотрицательными целыми числами, то добавить $(xs + r)$ к списку искомых делителей.

5. Если $a_i = 0$, то алгоритм заканчивает работу. Иначе осуществляется переход на шаг 2 к следующему значению $i = i + 1$.

ЗАДАНИЕ

Замечание: следует иметь в виду, что все представленные методы ищут только один делитель n. Поэтому необходимо примерять метод несколько раз, пока не получится полное разложение числа на простые множители.

I. Реализовать приложение, удовлетворяющее следующим требованиям.

1. Во входном файле хранятся входные данные, необходимые для работы программы (например, подлежащее факторизации число).

2. Программа проверяет заданное число на простоту с помощью теста, реализованного в лабораторной работе № 6. Если оно является простым, то процедура факторизации не выполняется.

3. Программа находит разложение заданного числа на произведение простых множителей.

4. Программа выдает список простых делителей заданного числа с указанием степени, с которой они входят в разложение числа, время и количество итераций основного цикла, потребовавшихся для разложения.

II. С помощью реализованного приложения выполнить следующие задания.

1. Протестировать правильность работы приложения на числах разной длины.

2. Сделать выводы о проделанной работе.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Отчет должен содержать:

- титульный лист (обязательно);
- задание на лабораторную работу (обязательно);
- описание метода решения заданий;
- описание разработанного программного средства;
- описание проведенных исследований (обязательно);
- программный код, написанный непосредственно студентами (обязательно);
- тестирование программы.

Отчет не должен содержать орфографических, пунктуационных и смысловых ошибок.

Все его разделы должны быть выдержаны в едином стиле оформления.

КРИТЕРИИ ОЦЕНИВАНИЯ КАЧЕСТВА РАБОТЫ

1. Графический интерфейс пользователя:
   1 – приложение имеет графический интерфейс пользователя;
   0 – приложение имеет интерфейс командной строки.
2. Выполнение требований к оформлению отчета:
   1 – отчет удовлетворяет всем требованиям;
   0 – отчет не удовлетворяет всем требованиям, но содержит обязательные разделы;
   Л.р. не принимается – в отчете нет хотя бы одного обязательного раздела.
3. Обработка ошибок:
   1 – все возможные ошибки и нестандартные ситуации (например неудачная попытка открытия файла) обрабатываются программой, которая выдает соответствующее сообщение;
   0 – не все возможные ошибки обрабатываются программой.

4. Применение принципов структурного программирования:

1 – все повторяющиеся либо логически целостные фрагменты программы выделены в качестве функций; работа каждой функции полностью определяется ее параметрами (т. е. не используются глобальные переменные, все данные, нужные функции для работы, передаются ей через параметры); программа позволяет без перекомпиляции изменять все параметры, от которых зависит ее работа; в тексте программы отсутствуют числовые константы (все необходимые константы объявляются как поименованные);

0 – иначе (не выполняется что-либо из перечисленного).

5. Наличие комментариев в тексте программы:

1 – комментариев достаточно для документирования исходных кодов;

0 – комментариев недостаточно.

6. Глубина понимания материала лабораторной работы каждым членом бригады:

1 – быстрые и правильные ответы на все вопросы;

0 – не на все вопросы ответы правильные и быстрые;

Л.р. не принимается – на половину вопросов ответы неправильные.

ВАРИАНТЫ

1. Метод Ферма.
2. (p – 1)-факторизация Полларда.
3. Метод Ленстры.
4. Метод Ферма (без операций деления).
5. Метод Лемана.
6. Метод Ферма.
7. Метод ρ-Полларда.
8. (p – 1)-факторизация Полларда.
9. Метод Ленстры.
10. Метод ρ-Полларда.
11. (p – 1)-факторизация Полларда.
12. Метод Ленстры.
13. Метод Ферма.
14. Метод Лемана.
15. Метод Ферма (без операций деления).

КОНТРОЛЬНЫЕ ВОПРОСЫ

I. Первая часть защиты

1. Метод факторизации Ферма.
2. Метод факторизации ρ-Полларда.
3. (p–1)-факторизация Полларда.
4. Метод Лемана.
5. Метод Ленстры.

II. Вторая часть защиты

Для заданного n найти все первообразные корни:

1. $n = 6$.
2. $n = 7$.
3. $n = 9$.
4. $n = 11$.
5. $n = 13$.
6. $n = 19$.

III. Третья часть защиты

Решить задачу дискретного логарифма методом согласования:

1. $5^n \equiv 2 \pmod 3$.
2. $3^n \equiv 9 \pmod 7$.
3. $3^n \equiv 6 \pmod 7$.
4. $5^n \equiv 4 \pmod 9$.
5. $4^n \equiv 1 \pmod{11}$.
6. $2^n \equiv 1 \pmod 9$.