РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

Отчёт по лабораторной работе №6

Дисциплина: *Математические основы защиты информации и информационной безопасности*

Студент: Майорова О.А., НФИмд-02-21
Преподаватель: д.ф.-м.н. Кулябов Д.С.

Москва 2021

Содержание

List of Figures

List of Tables

1 Цель работы

Цель: Ознакомиться с задачей разложения составного числа на множители.

2 Задание

Программно реализовать ρ-метод Полларда.

3 Теоретическое введение

Факторизацией натурального числа называется его разложение в произведение простых множителей. Существование и единственность (с точностью до порядка следования множителей) такого разложения следует из основной теоремы арифметики [1]. Согласно основной теореме арифметики любое положительное целое число больше единицы может быть уникально записано в следующей главной форме разложения на множители, где $p_1, p_2, ..., p_k$ — простые числа и $e_1, e_2, ..., e_k$ — положительные целые числа:

$n = p_1^{e1} \times p_2^{e_2} \times \cdots \times p_k^{e_k}$.

Есть непосредственные приложения разложения на множители, такие как вычисление наибольшего общего делителя и наименьшего общего множителя [2]. Предположение о том, что для больших чисел задача факторизации является вычислительно сложной, лежит в основе широко используемых алгоритмов (например, RSA). Задача поиска эффективных способов разложения целых чисел на множители интересовала математиков с давних времён, особенно специалистов в области теории чисел. Как правило, на вход таких алгоритмов подаётся число $n \in N$, которое необходимо факторизовать, состоящее из $N = (\log_2 n) + 1$ символов, если n представлено в двоичном виде [1].

В 1975 г. Джон М. Поллард разработал метод для разложения на множители, который базируется на следующих положениях:

1. Предположим, что есть два целых числа, x_1 и x_2, таких, что p делит $x_1 – x_2$, но эта разность не делится на n.

2. Может быть доказано, что $p =$ НОД$(x_1 – x_2, n)$. Поскольку p делит $x_1 – x_2$,

можно записать, что $x_1 – x_2 = q \times p$. Но поскольку n не делит $x_1 – x_2$, очевидно, что q не делится на n. Это означает, что НОД$(x_1 – x_2, n)$ является либо 1, либо сомножителем.

Следующий алгоритм повторно выбирает x_1 и x_2, пока не находит соответствующую пару:

1. Выберите x_1 — малое случайное целое число, называемое первоисточником.

2. Используйте функцию, чтобы вычислить x_2, такую, чтобы n не делило $x_1 – x_2$. Функция, которая может быть применена, — это $x_2 = f(x_1) = x_1^2 + a$ (a обычно выбирается как 1).

3. Вычислить НОД$(x_1 – x_2, n)$. Если это не 1, результат – сомножитель. Алгоритм останавливается. Если это 1, то происходит возвращение, чтобы повторить процесс с x_1. Теперь мы вычисляем x_3. Заметим, что в следующем раунде мы начинаем с x_3 и так далее. Если мы перечислим значения нескольких x, используя ρ-алгоритм Полларда, мы увидим, что дуга значений в конечном счете повторяется, создавая форму, подобную греческой букве ρ.

Чтобы уменьшить число итераций, алгоритм был немного изменен. Он начинается с пары (x_0, x_0), и итеративно вычисляет (x_1, x_2), (x_2, x_4), (x_3, x_6), ..., (x_i, x_{2i}), используя равенство $x_{i+1} = f(x_i)$. В каждой итерации мы применяем функцию $f(x_i)$ (начиная с шага 2). При этом вычисление идут следующим образом: в паре вычисляется один раз первый элемент и дважды вычисляется второй элемент [2].

4 Выполнение лабораторной работы

Для выполнения лабораторной работы был выбран язык Python. Далее реализуем представленный алгоритм в виде функции в соответствии псевдокоду из задания к лабораторной работе.

Сначала реализуем функцию, обладающую сжимающими свойствами:

```
def f(x, n):
    return (x**2 + 5) % n
```

Так как для ρ-метода Полларда необходимо вычисление наибольшего общего делителя, используем чуть изменённую функцию, реализующую алгоритм Евклида, из лабораторной работы 4:

```
def Euclid(a, b):
    rp = a
    rc = b
    rn = 1
    while rn != 0:
        rn = rp % rc
        d = rc
        rp = rc
        rc = rn

    return d
```

Наконец, реализуем ρ-метод Полларда:

```
def Pollard(c, n):
    a = c
    b = c
    while True:
        a = f(a, n) % n
        b = f(f(b, n), n) % n
        d = Euclid(a-b, n)
        if 1 < d < n:
            return d

        if d == n:
            return 'Делитель не найден'
```

Результатом запуска функции будет рис. 4.1.

```
1  Pollard(1, 1359331)
```

```
1181
```

Figure 4.1: Проверка функции

Можно видеть, что был получен верный результат, и функция работает корректно.

5 Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы. Было осуществлено знакомство с задачей разложения составного числа на два нетривиальных сомножителя. Также была получена реализация на языке Python ρ-метода Полларда.

Список литературы

1. Факторизация целых чисел [Электронный ресурс]. Википедия: Свободная энциклопедия, 2021. URL: https://ru.wikipedia.org/w/index.php?title=Факторизация_целых_чисел&oldid=117143943.
2. Лекция 12: Простые числа [Электронный ресурс]. НОУ «ИНТУИТ», 2021. URL: https://intuit.ru/studies/courses/552/408/lecture/9368.