**ИССЛЕДОВАНИЕ МЕТОДОВ ДИСКРЕТНОГО ЛОГАРИФМИРОВАНИЯ**

***Гусев В.Е.***

*Научный руководитель - канд. техн. наук Мубараков Б.Г.*

*Казанский (Приволжский) федеральный университет, Институт Вычислительной математики и информационных технологий*

*viegusev@stud.kpfu.ru*

В процессе практики были изучены и реализованы модификации методов дискретного логарифмирования на языке программирования C# на .NET8 в Windows Forms. Также для тестирования данных алгоритмов был реализован генератор параметров Диффи-Хеллмана, возведение числа в степень по модулю, замер времени выполнения алгоритма и количество потраченной памяти на выполнение алгоритма.

Были реализованы модификации экспоненциальных алгоритмов дискретного логарифмирования: алгоритм Шенкса, алгоритм Полига-Хеллмана, ро-метод Полларда, а также модификации субэкспоненциальных алгоритмов дискретного логарифмирования: алгоритм Адлемана, алгоритм COS, решето числового поля. Для разработки всех алгоритмов были реализованы вспомогательные математические функции.

Была реализована функция быстрого возведения в степень по модулю. Возведение в степень по модулю — это операция над натуральными числами возведения в степень, выполняемая по модулю. Находит применение в информатике, особенно, в области криптографии с открытым ключом. Возведение в степень по модулю — это вычисление остатка от деления натурального числа (основание), возведенного в степень (показатель степени), на натуральное число (модуль). Обозначается: .

Для проверки чисел на простоту был реализован тест Миллера-Рабина. Данный тест является вероятностным полиномиальным тестом простоты. Тест Миллера-Рабина, наряду с тестом Ферма и тестом Соловея-Штрассена, позволяет эффективно определить, является ли данное число составным. Однако, с его помощью нельзя строго доказать простоту числа. Тем не менее тест Миллера-Рабина часто используется в криптографии для получения больших случайных простых чисел.

Была реализована модификация экспоненциального алгоритма «Шаг младенца — шаг великана» — в теории групп детерминированный алгоритм дискретного логарифмирования в мультипликативной группе кольца вычетов по модулю простого числа. Был предложен советским математиком Александром Гельфондом в 1962 году и Дэниелом Шенксом в 1972 году. Метод теоретически упрощает решение задачи дискретного логарифмирования, на вычислительной сложности которой построены многие криптосистемы с открытым ключом. Относится к методам встречи посередине. Это был один из первых методов, который показал, что задача вычисления дискретного логарифма может быть решена значительно быстрее, чем методом перебора. Идея алгоритма состоит в выборе оптимального соотношения времени и памяти, а именно в усовершенствованном поиске показателя степени.

Была реализована модификация алгоритма «Шаг младенца — шаг великана», состоящая в распараллеливании 2 и 3 шага алгоритма. На 2 шаге алгоритма параллельно вычисляются два ряда чисел. На 3 шаге был сделан параллельный поиск результата с начала и с конца ряда. Данная модификация предположительно должна увеличить скорость работу с рядами чисел.

Для проверки работы модифицированного алгоритма «Шаг младенца — шаг великана» были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 81 мс и затратил памяти 3947976 байт, а модифицированный за 120 мс и затратил 2987840 байт.

Была реализована модификация детерминированного алгоритма дискретного логарифмирования с экспоненциальной сложностью Полига-Хеллмана в кольце вычетов по модулю простого числа. Одной из особенностей алгоритма является то, что для простых чисел специального вида можно находить дискретный логарифм за полиномиальное время. Данный алгоритм был придуман американским математиком Роландом Сильвером, но впервые был опубликован другими двумя американскими математиками Стивеном Полигом и Мартином Хеллманом в 1978 году в статье «An improved algorithm for computing logarithms over GF(p) and its cryptographic significance», которые независимо от Роланда Сильвера разработали данный алгоритм.

Была реализована модификация алгоритма Полига-Хеллмана, состоящая в том, что на 1 шаге алгоритма число было разложено на простые множители и данные простые множители были возведены в свои степени, чтобы на 2 шаге была составлена таблица из единичных значений без степеней. Данная модификация предположительно должна будет уменьшить расход памяти при составлении таблицы на 2 шаге и ускорить вычисления на 3 и 4 шаге алгоритма.

Для проверки работы модифицированного алгоритма Полига-Хеллмана были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 7 мс и затратил памяти 2426080 байт, а модифицированный за 6 мс и затратил 3806344 байт.

Была реализована модификация экспоненциального алгоритма дискретного логарифмирования ро-метод Полларда для факторизации (разложения на множители) целых чисел. Данный алгоритм основывается на алгоритме Флойда поиска длины цикла в последовательности и некоторых следствиях из парадокса дней рождения. Ро-метод Полларда строит числовую последовательность, элементы которой образуют цикл, начиная с некоторого номера , что может быть проиллюстрировано, расположением чисел в виде греческой буквы ρ, что послужило названием семейству алгоритмов.

Была реализована модификация алгоритма ро-метод Полларда, состоящая в том, что на 4 шаге алгоритма увеличилась степень вычисляемого . При вычислении x степень полинома увеличилась до 3. Данная модификация предположительно увеличит скорость вычисления алгоритма в цикле на 3 шаге, но увеличит количество затрачиваемой памяти.

Для проверки работы модифицированного алгоритма ро-метод Полларда было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 1 мс и затратил памяти 7968 байт, а модифицированный за 1 мс и затратил 8224 байт.

Была реализована модификация алгоритма Адлемана, который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа. Алгоритм был предложен Леонардом Максом Адлеманом в 1979 году. Леонард Макс Адлеман — американский учёный-теоретик в области компьютерных наук, профессор компьютерных наук и молекулярной биологии в Университете Южной Калифорнии. Он известен как соавтор системы шифрования RSA и ДНК-вычислений. RSA широко используется в приложениях компьютерной безопасности, включая протокол HTTPS.

Была реализована модификация алгоритма Адлемана, состоящая в том, что на 1 шаге алгоритма был изменён показатель степени при вычислении числа , тем самым повысив факторную базу. Данная модификация предположительно должна увеличить таблицу на 2 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма Адлемана были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 268 мс и затратил памяти 48040 байт, а модифицированный за 11612 мс и затратил 532072 байт.

Была реализована модификация алгоритма COS (Копперсмит, Одлыжко, Шреппель), который является первым субэкспоненциальным алгоритмом дискретного логарифмирования в кольце вычетов по модулю простого числа. В 1986 г. Копперсмит, Одлыжко и Шреппель предложили алгоритм дискретного логарифмирования с эвристической оценкой сложности арифметических операций.

Была реализована модификация алгоритма COS, состоящая в том, что на 2 шаге был увеличен наименьший вычет, добавив значение , чтобы увеличить разложение чисел при формировании СЛАУ. Данная модификация предположительно должна увеличить таблицу на 3 шаге для большего перебора результата алгоритма.

Для проверки работы модифицированного алгоритма COS были сгенерированы параметры Диффи-Хеллмана: . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 129 мс и затратил памяти 907296 байт, а модифицированный за 81 мс и затратил 4747168 байт.

Была реализована модификация алгоритма решето числового поля, который является методом факторизации целых чисел. Метод решета числового поля (как специальный, так и общий) можно представить как усовершенствование более простого метода — метода рационального решета либо метода квадратичного решета. Подобные им алгоритмы требуют нахождения гладких чисел порядка . Размер этих чисел экспоненциально растёт с ростом . Метод решета числового поля, в свою очередь, требует нахождения гладких чисел субэкспоненциального относительно размера. Благодаря тому, что эти числа меньше, вероятность того, что число такого размера окажется гладким, выше, что и является причиной эффективности метода решета числового поля. Для достижения ускорения вычислений в рамках метода проводятся в числовых полях, что усложняет алгоритм, по сравнению с более простым рациональным решетом.

Была реализована модификация алгоритма решето числового поля, состоящая в том, что на 2 шаге алгоритма выбирается степень неприводимого многочлена, равное количество байт входного числа . Данная модификация предположительно поможет эффективно выбирать не случайным образом степень полинома для дальнейших вычислений и повысит скорость вычисления алгоритма.

Для проверки работы модифицированного алгоритма решето числового поля было сгенерировано число . В результате работы модифицированного алгоритма был корректно вычислен результат . Начальный алгоритм выполнился за 540 мс и затратил памяти 2014208 байт, а модифицированный за 235 мс и затратил 792064 байт.