**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ»**

**Факультет безопасности информационных технологий**

**Дисциплина:**

**“Алгоритмы и структуры данных”**

**Отчет о работе на тему**

***“*Метод квадратичного решета разложения целых чисел на множители, метод сопряженных градиентов решения СЛАУ с разреженной матрицей*”***

**Выполнил:**

Студент гр. N3249

Шарифуллин Ильдан Айдарович



**Проверил:**

Доцент, доктор технических наук

Трифонов Петр Владимирович

Санкт-Петербург

2022г.

Я провел исследование скорости работы написанной программы, реализующей метод квадратичного решета. По итогу этого исследования получился следующий график зависимости времени работы от размерности числа:

На нем оранжевой линией отображены результаты тестов с методом сопряженных градиентов, синей – с методом Гаусса.

Безусловно, для метода сопряженных градиентов скорость работы должна быть выше, нежели для метода Гаусса. В своей программе я допустил ряд упрощений алгоритма, что и снизило скорость работы. В частности, разряженную матрицу я хранил не в виде списка индексов, а в виде матрицы с единицами и нулями. Конечно, это сильно снижает скорость работы алгоритма.

В ходе написания программы я впервые поработал с программной реализацией арифметики в поле Галуа. Её реализация не показалась мне сложной, однако и для нее нужно было постараться написать все оптимально.

Также во время тестирования я столкнулся с проблемой того, что программа отрабатывает за разное время с одними и теми же тестами. Причины этому две:

1. Метод сопряженных градиентов является итерационным. Мы не можем заранее сказать, за какое время точно найдется решение для определенной СЛАУ.
2. В процессе итераций метода сопряженных градиентов берется случайное начальное приближение, которое может оказаться “неудачным” для нахождения решения. В таком случае мы брали следующее случайное начальное приближение. Из-за этой доли случайности, время и разнилось порой на довольно большие величины.

Для того, чтобы максимально снизить долю влияния случайного выбора начального приближения на результат работы, каждый тест я проводил от 10 до 20 раз, далее отбирал значения, находящиеся в диапазоне Ṡ ± 3σ (где Ṡ - среднее значение, а σ – среднеквадратичное отклонение), после чего строил график по усредненному значению оставшихся величин.

И, конечно, в ходе работы я детально проработал приведенные в реферате методы. Перед написанием программы тесты алгоритмов я проводил в программе Maple, что позволило мне довольно быстро написать саму программу на языке C++.