

# Лабораторная работа № 6 по курсу дискретного анализа: Динамическое программирование

Выполнила студентка группы 08-308 МАИ *Шевлякова София*.

## Условие

Необходимо разработать программную библиотеку на языке C или C++, реализующую простейшие арифметические действия и проверку условий над целыми неотрицательными числами. На основании этой библиотеки, нужно составить программу, выполняющую вычисления над парами десятичных чисел и выводящую результат на стандартный файл вывода.

Список арифметических операций:

- Сложение (+).
- Вычитание (-).
- Умножение (\*).
- Возведение в степень ( $\wedge$ ).
- Равно (=).

В случае возникновения переполнения в результате вычислений, попытки вычесть из меньшего числа большее, деления на ноль или возведения нуля в нулевую степень, программа должна вывести на экран строку Error.

Список условий:

- Больше ( $>$ ).
- Меньше ( $<$ ).
- Равно (=).

В случае выполнения условия, программа должна вывести на экран строку true, в противном случае – false.

**Формат ввода:** входной файл состоит из последовательностей заданий, каждое задание состоит из трех строк:

- Первый операнд операции.
- Второй операнд операции.
- Символ арифметической операции или проверки условия (+, -, \*,  $\wedge$ , /, >, <, =).

Числа, поступающие на вход программе, могут иметь «ведущие нули».

**Формат вывода:** для каждого задания из входного файла нужно распечатать результат на отдельной строке в выходном файле:

- Числовой результат для арифметических операций.
- Строку Error в случае возникновения ошибки при выполнении арифметической операции.
- Строку true или false при выполнении проверки условия.

В выходных данных вывод чисел должен быть нормализован, то есть не содержать в себе «ведущих» нулей.

## Метод решения

Требуется реализовать алгоритм для работы с длинными числами. Можно работать с числами, как со строками, то есть в системе счисления 10, но, если использовать систему счисления больше, можно ускорить программу в несколько раз, поэтому я использовала систему счисления с основанием  $10^3$ . Будем хранить массив троек числа. Вводить число нужно будет строкой, поэтому нужно уметь преобразовывать строку в массив цифр. При печати числа на экран нужно помнить о недостающих нулях в цифрах числа и выводить их.

Складывать, вычитать, умножать и делить числа будем школьным способом в столбик — это наиболее простой алгоритм. Сравнение чисел не отличается от лексикографического сравнения строк. Для возведения в степень воспользовалась бинарным возведением в степень. Для проверки ошибок в делении и возведении в степень написала метод IsZero().

## Описание программы

В данной программе содержится один файл main.cpp, в котором реализован класс BigInteger.

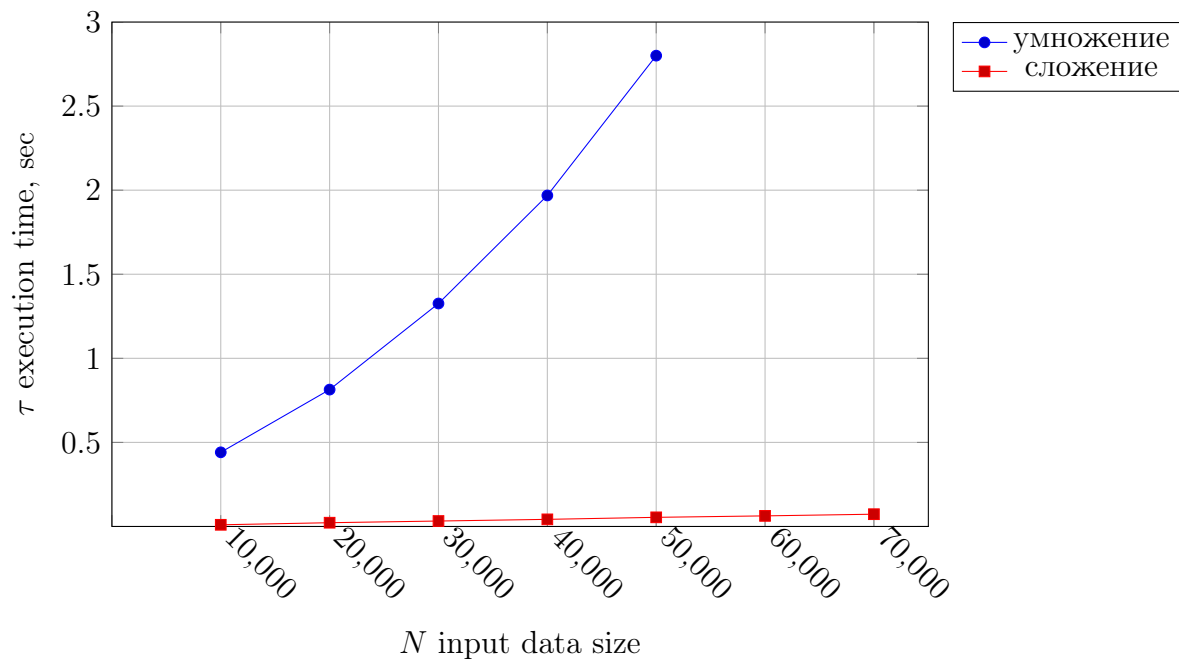
Основные функции:

- Функции сравнения чисел ищут первую различающую цифру и по ней делают вывод. Функции арифметических операций основаны на школьном алгоритме вычисления в столбик. Для ускорения программы при делении находим очередную цифру ответа бинарным поиском.
- Вспомогательные функции каррирования для положительных и отрицательных чисел отдельно, и функция удаления ведущих нулей. При удалении ведущих нулей для более эффективной реализации (чтобы не удалять из начала массива) один раз развернем числа, удалим все его нули из конца, сложность такой операции константная, и обратно перевернем число. В функциях каррирования проверяем, что число не выходит за пределы допустимых значений, и, если требуется, выполняем перенос через разряд.

## Дневник отладки

№	Описание ошибки	Способ устранения
1	WA4: была неправильно написана формула для вычитания.	Исправила формулу на правильную, поменяла индексы.

## Тест производительности



Не сложно оценить, и видно по графику, что сложность таких операций как считывание, сложение, каррирование, удаление ведущих нулей, печать числа линейная:  $O(n)$ , где  $n$  — длина входного числа.

Не более сложным является и умножение, в соответствии умножения в столбик мы одно число проходим столько раз, сколько разрядов во втором. Тоже видно по графику, что сложность является квадратичной относительно длины входного числа:  $O(n^2)$ .

## Выводы

Проделав лабораторную работу, я реализовала длинную арифметику в системе счисления  $10^3$ . Самыми сложными в реализации для меня оказались операции деления.

Длинная арифметика может применяться в криптографии. Большинство систем и шифрования данных используют целочисленную арифметику по модулю  $m$ , где  $m$  — очень большое натуральное число, не обязательно простое.