Лабораторная работа № 6 по курсу дискретного анализа: Калькулятор. Длинная арифметика

Выполнил студент группы 08-208 МАИ *Куликов Алексей*.

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 (^).
- Деление (/).

В случае возникновения переполнения в результате вычислений, попытки вычесть из меньшего числа большее, деления на ноль или возведении 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.

Метод решения

Простейшие операции сложения и вычитания полность повторяют процедуру сложения или вычитания «в столбик» на листе бумаги. Т.е. начиная с младших разрядов складываем или вычитаем разряды, где надо запоминаем избыток (для сложения), или «занимаем» (для деления) и учитываем это во время операции над следующим разрядом. Для сложения, если после прохода по всем разрядом остался избыток, то добавляем еще один разряд и избыток записываем туда.

Чуть сложнее операция умножения, и, кроме того, она слегка отличается от процедуры умножения «на листочке». Сдесь суммиррование по результатам перемножений

разрядов происходит сразу, и, если возникает избыток, учитываем его на следующем этапе. Так же, если после прохода по всем разрядам, остался избыток, то приписываем его в конец числа-результата.

Деление осуществляется по-сути так же, как «в столбик», только здесь необходим алгоритм для угадывания числа, на которое надо домножить делитель, чтобы оно вместилось в оставшееся от делителя к текущему этапу. Для этого здесь использован бинарный поиск. В остальном процедура деления та же.

Возведение в степень реализовано бинарно.

Операции сравнения предельно просты. Просто идем от младших разрядов к старшим, до первого несовпадения и в зависимости от отношения этих двух результатов возвращаем результат. Так же не имеет смысла сравнивать числа разной длины, поэтому уже из этого можно делать вывод об отношении чисел.

Описание программы

Программа состоит из единственного исходного файла. В нем определен класс `TBigInt`, реализующий арифметические операции над длинными числами.

Имеется парочка конструкторов (из строки или 16-разрядного инта), опрератор вывода, всевозможные арифметические операции: сложение, вычитание, умножение, целочисленное деление, возведение в степень (длинное-длинное) и деление длинного на короткое, и возведение в короткую степень (это сделано скорее для ускорения работы длинных операторов). Все операторы реализованы вышеописанным образом.

Дневник отладки

Особых проблем не возникало, программа прошла проверку с первого раза.

Тест производительности

Таблица 1: Скорость выполнения арифметических операций

Тип операции	**Время работы (мкс.)**
Сложение	3
Вычитание	3
Умножение	12
Деление	90
Возведение в степень	44379

Сложение, вычитание, умножение и деление измерялись для чисел порядка 10^{94}. Возведение степень измерялись с основанием порядка 10^5 и показателем степение порядка 10^4.

Выводы

Длинная арифметика может применяться в криптографии. Большинство систем подписывания и шифрования данных используют целочисленную арифметику по модулю m, где m — очень большое натуральное число, не обязательно простое. Например, при реализации метода шифрования RSA, криптосистемы Рабина или схемы Эль-Гамаля требуется обеспечить точность результатов умножения и возведения в степень порядка 10^{309} (сам пока не сталкивался, но верю Википедии на слово).

В целом, большой сложности реализация базовой длинной арифметики над целыми числами без каких-то хитрых оптимизаций сложности не представляет. Более быстрые же реализации требуют более сложных алгоритмов, которых, вероятно, существует даже несколько. К счастью их реализовывать пока не пришлось.