Лабораторная работа № 6 по курсу дискретного анализа: Калькулятор

Выполнил студент группы М8О-308Б-22 МАИ *Немкова Анастасия.*

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 (=).

Замечание: при реализации деления можно ограничить делитель цифрой внтуреннего представления «длинных» чисел, в этом случае максимальная оценка, которую можно получить за лабораторную работу, будет ограничена оценкой 3 («удовлетворительно»).

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

Длинные числа храним в виде вектора его цифр в порядке убывания разрядов.

Сложение и вычитание выполняются в столбик. При сложении двух чисел каждый блок складывается отдельно, начиная с младших разрядов. Если сумма блока превышает 10000, сохраняется "перенос"в следующий разряд. В конце, если остался перенос, он добавляется как новый блок в результат. При вычитании проверяем, что уменьшаемое больше вычитаемого и также выполняем его по блокам, начиная с младших разрядов. Если текущий блок меньше, чем блок вычитаемого числа, необходимо "заимствовать"из следующего разряда, увеличивая его на 10000. Заимствование учитывается при вычислении разности текущего блока. Асимптотика обеих операций - O(n).

Для сравнения чисел сравниваем их длины, если они одинаковы, то сравниваем разряды. Эти операции также выполняются за O(n).

Умножение выполняется по аналогии с обычным умножением в столбик. Каждый блок первого числа умножается на каждый блок второго числа, и результаты складываются, учитывая позицию разряда. Если результат превышает 10000, сохраняется перенос для добавления в более старшие разряды. Умножение выполняется за O(n * m).

Алгоритм деления использует бинарный поиск для нахождения частного. От текущего остатка (в котором постепенно добавляются блоки делимого) ищется максимальный множитель делителя, который помещается в текущий блок результата. Остаток обновляется путем вычитания произведения делителя на найденный множитель. Сложность операции - O(n * m * log(a)), a - основание системы счисления.

Возведение в степень реализуется с использованием метода быстрого возведения в степень. Если показатель степени четный, число возводится в квадрат, и степень делится на два. Если степень нечетная, результат умножается на основание, а степень уменьшается на единицу. Сложность операции - O(nlog(m)).

Описание программы

Для работы с длинными целыми числами был создан класс TBigInt. Он поддерживает основные арифметические операции, такие как сложение, вычитание, умножение, деление и возведение в степень. Также реализованы операторы сравнения и ввод-вывод через стандартные потоки.

- TBigInt(const std::string str) - конструктор числа из строки
- void DeleteZeros() - удаление лидирующих нулей
- friend TBigInt operator+ (const TBigInt num1, const TBigInt num2) - перегрузка опрератора сложения
- friend TBigInt operator- (const TBigInt num1, const TBigInt num2) - перегрузка опреатора вычитания

- friend TBigInt operator* (const TBigInt num1, const TBigInt num2) - перегрузка оператора умножения
- friend TBigInt operator/ (const TBigInt num1, const TBigInt num2) - перегрузка оператора деления
- friend TBigInt operator∧ (TBigInt num1, TBigInt num2) - перегрузка оператора возведения в степень
- friend bool operator== (const TBigInt num1, const TBigInt num2) - перегрузка оператора равно
- friend bool operator> (const TBigInt num1, const TBigInt num2) - перегрузка оператора больше
- friend bool operator< (const TBigInt num1, const TBigInt num2) - перегрузка оператора меньше
- friend std::istream operator» (std::istream in,TBigInt num) - перегрузка оператора ввода
- friend std::ostream operator« (std::ostream out, const TBigInt num) - перегрузка оператора вывода

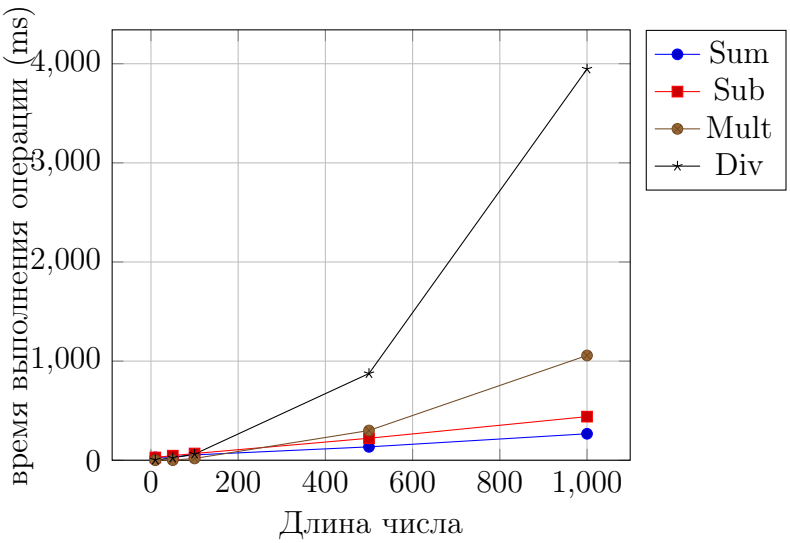
Дневник отладки

1. 2 окт 2024, 20:09:54 WA на 3 тесте

   Неверно реализовано сложение. Решение: если после сложения всех разрядов остался непустой перенос, он добавляется как новый старший разряд числа

Тест производительности

Для измерения производительсти протестируем выполнение арифметических операций: сложения, вычитания и умножения на входных данных различной длины. Возьмем длины чисел : 10, 50, 100, 500, 1000 и будем замерять группами по 100 операций.



Выводы

В ходе лабораторной работы была реализована библиотека для работы с большими целыми числами, которая поддерживает такие основные арифметические операции, как сложение, вычитание, умножение, деление и возведение в степень. Реализации данных операций не единственны, существуют альтернативные алгоритмы, такие как умножение Карацубы, которое ускоряет процесс умножения за счет рекурсивного деления числа на части. Существуют также более продвинутые алгоритмы для деления, такие как метод Ньютона, который может ускорить вычисления за счет более эффективного поиска частного.