

Лабораторная работа № 6 по курсу дискретного анализа: калькулятор

Выполнил студент группы М8о-207Б-18 МАИ *Токарев Никита*.

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.

Количество десятичных разрядов целых чисел не превышает 100000. Основание выбранной системы счисления для внутреннего представления «длинных» чисел должно быть не меньше 10000.

Метод решения

Для начала нужно определить подходящий base для моей системы счисления. В моей системе счисления $base = 1000000$, то есть один разряд моего реализуемого числа принадлежит промежутку $0 \leq x < 1000000$. Сложность операции деления и возведения в степень будут указаны пренебрегая сложностью внутренних операций.

1. Операция сложения: Данную операция реализуется с помощью прохода по двум числам от младшего разряда к старшему с последующим сложением. Сложение реализуется по правилу сложения столбиком. Сложность $O(n)$, где n - размер максимального числа среди двух данных.

2. Операция вычитания: Данная операция практически аналогична сложению, только происходит вычитание. Также если разность двух разрядов числа отрицательна, то необходимо взять 1 с более высшего разряда. Вычитание также реализуется по правилу вычитания столбиком. Сложность $O(n)$, где n - размер максимального числа среди двух данных.
3. Операция умножения: Данная операция аналогична операции умножения в столбик. Для начала я определяю максимальный размер числа `result`, которое может получиться в процессе умножения. `result` я инициализирую нулями, а затем последовательно от младшего разряда к старшему умножаю разряды первого множителя на второй множитель. Сложность $O(n * m)$, где n - первого множителя, m - второго множителя.
4. Операция деления: Логика схожа с делением уголком. Начиная со старшего разряда получаю делимое и нахожу частное, с помощью бинарного поиска, при делении на делитель. Затем же я отнимаю, полученное частное умноженное на делитель, от делимого. Сложность $O(n * \log m)$, где n число разрядов делителя, а $\log m$ - сложность бинарного поиска.
5. Операция возведения в степень: Реализовано бинарное возведение в степень. Сложность $O(\log n)$, где n степень числа.

Описание программы

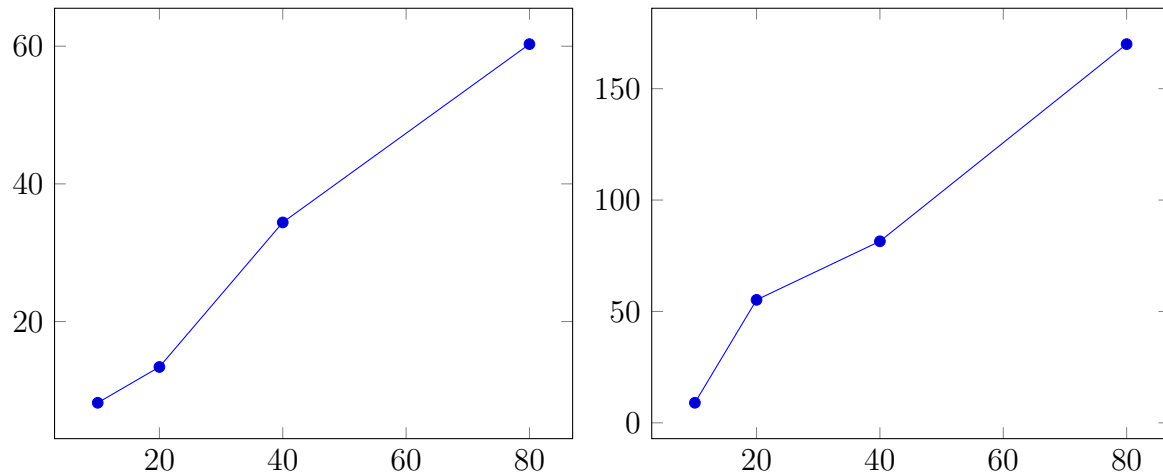
В данной программе для хранения числа я использовал шаблон `std::vector`, а также тип `long long`. В классе `BigInt` перегружены такие операторы: деления, умножения, вычитания, сложения, а также сравнения. Также в данной работе мне понадобилось три конструктора: конструктор по умолчанию, конструктор инициализирующий вектор нулями, а также конструктор принимающий на вход строку. Операцию возведения в степень вызывается как функция, где на вход подается число типа `int`, являющиеся степенью числа. Так же хотелось бы отметить, что также перегружен оператор вывода (`>>`). Вывод происходит таким образом, что число печатается слева-направо и также в процессе вывода разряды числа дополняются ведущими нулями.

Дневник отладки

1. Неправильный ответ в 4 тесте: происходило переполнение типа `int` в операции умножения.
2. Неправильный ответ в 10 тесте: некорректно было реализовано возведение в степень.
3. Ошибка выполнения в 10 тесте: возведение в степень было реализовано с большей асимптотической сложностью, таким образом происходило превышение `std::vector` или получалось очень большое число.

Тест производительности

Тесты представляют из себя набор из чисел, где размер числа принадлежит промежутку от $[1, 10000]$. Тут я продемонстрирую работу операции деления и возведения в степень. Хотелось бы отметить, что числа будут возводиться в степень не больше 10. Ось x - количество операций. Ось y - время в секундах.



Слева представлен график тестирования операции деления, а справа возведения в степень.

Выводы

В некоторых языках программирования логика длинной арифметики уже встроена в сам язык программирования. В ходе данной работы я реализовал простейший калькулятор. Конечно же есть возможность улучшить калькулятор, добавив возможность работы с отрицательными числами и также оптимизировав некоторые алгоритмы. Также умножение было реализовано без использования алгоритма Карацубы, который, как мне известно, заметно быстрее работает при очень больших числах.