

Лабораторная работа № 6 по курсу дискретного анализа: жадные алгоритмы

Выполнил студент группы М8О-208Б-18 МАИ *Коростелев Дмитрий Васильевич*.

Задание

Необходимо разработать программную библиотеку на языке С или С++, 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).

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.

Формат входных данных

Необходимо разработать программную библиотеку на языке С или С++, 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).

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.

Формат результата

Для каждого задания из выходного 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.

Строки `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 класс, который будет поддерживать различные операции над длинными числами. Наполнение класса – это база числа и контейнер, где содержаться части числа меньшие заданной базы, при чем, чтобы интерпретировать сохраненные части числа в обычное число, нужно рассматривать массива с конца до начала. Класс `TLongNumber` поддерживает следующие функции – `Clear`, `Print`, `Sum`, `Sub`, `Mult`, `Div`, `SmallDiv`, `Exp`, `SmallExp`.

`void TLongNumber::Clear()` – очищает контейнер.

`void TLongNumber::Print()` – выводит длинное число.

`int Cmp(const TLongNumber& a, const TLongNumber& b)` – сравнивает два числа, возвращает один, если `a` больше `b`, 0 – `a` равно `b`, -1 – `b` больше `a`. (Асимптотика – $O(\min(a.size(), b.size()))$)

`TLongNumber Sum(const TLongNumber& a, const TLongNumber& b)` – суммирует два числа, возвращает новое. Алгоритм точно такой же, как обычное сложение «столбиком». (Асимптотика – $O(\max(a.size(), b.size()))$)

`TLongNumber Sub(const TLongNumber& a, const TLongNumber& b)` – вычи-

тает из a b , при этом заведомо должно быть, что a больше b . Алгоритм – поразрядно выполняем вычитание сохраняя остаток. (Асимптотика – $O(\max(a.size(), b.size()))$)

TLongNumber Mult(const TLongNumber& a, const TLongNumber& b) – умножает a на b . Берем i и j разряд и помещаем сумму умножения этих разрядов и остаток, посчитанный на прошлом шаге, при этом, если результат больше базы, то вычитаем базу и остаток сохраняем. (Асимптотика – $O(a.size()*b.size())$)

TLongNumber Div(const TLongNumber& a, const TLongNumber& b) – делит a на b . В ходе выполнения алгоритма проходим по все разрядам числа a , формируя на каждом новом шаге новое число, добавляя новые разряды из числа a , далее подбирается частное посредством бинарного поиска и перемножения длинных чисел на простой int . Итоговая асимптотика $O(a.size()*b.size()*\log(Base))$

TLongNumber SmallExp(const TLongNumber& a, const TLongNumber& b) – возводит a в степень b – так как перемножение происходит не подряд, а половинами от текущего кол-ва b , то итоговая сложность – $O(a.size()*b.size()*\log(b))$

Отладка и проверка программы.

№	Название ошибки	Причина возникновения ошибки
1-6	Ошибка выполнения	Ошибка в делении
7-14	Неправильный ответ	Ошибка в делении
14-30	Неправильный ответ	Ошибка в возведении в степень
14-30	Превышено реальное время работы	Ошибка в возведении в степень

Вывод

Длинная арифметика позволяет реализовать числа любых размеров и вычисления над ними, однако реализация, некоторых, казалось бы простых операций может вызвать некоторую сложность.