Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №6 по курсу «Дискретный анализ»

Студент: И. Д. Черненко
Преподаватель: А. А. Кухтичев
Группа: М8О-306Б

Дата:
Оценка:
Подпись:

Москва, 2021

Лабораторная работа №6

Задача: 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.

1 Описание

Число хранится в векторе с базой длинного числа 1000000000. Элемент вектора - это одна цифра в 1000000000-ой системе счисления. Для удобства выполнения операций разряды числа хранятся в векторе в обратном порядке.

Сложение реализовано столбиком: начинаем с младших разрядов, складываем цифры, стоящие в этих разрядах, и прибавляем к сумме остаток от сложения предыдущих разрядов. Записываем остаток от деления этой суммы на базу в ответ, при этом новым остатком будет являться частное суммы и базы. Сложность операции сложения $O(max(n, m))$, где n - количество разрядов в первом числе, m - количество разрядов во втором числе.

Вычитание реализовано столбиком: начинаем с младших разрядов, вычитаем цифры, стоящие в этих разрядах, а также вычитаем цифру, которую занимали в предыдущем вычитании. Если разность получилась отрицательная, то прибавляем к полученной разности базу, тем самым занимаем единицу из следующего разряда. Вычитание не предполагает что в результате получится отрицательное число. Сложность операции вычитания $O(n)$, где n - количество разрядов в уменьшаемом числе.

Умножение реализовано столбиком: Для i-го разряда второго числа ($i = 0..m - 1$) происходит умножение на каждый разряд j первого числа ($j = 0..n - 1$). Полученное произведение i-го разряда второго числа и j-го разряда второго числа складывается с остатком предыдущего умножения. $i+j$-й разряд результирующего вектора увеличивается на частное от деления этой суммы на базу. А новым остатком будет являться результат деления суммы на базу. Сложность операции умножения столбиком равна $O(n * m)$, где n - количество разрядов в первом числе, m - количество разрядов во втором числе.

Деление реализовано уголком: На каждой итерации имеем текущее значение, которое пытаемся уменьшить на максимально большое количество раз делимым, это количество вычисляется бинарным поиском. Сложность операции деления: $O(n * m * log(BASE)$.

Возведение в степень имеет сложность $O(n^2 * m^2 * log(m))$,где n - число, возводимое в степень, m - степень.

Операции сравнения осуществляются поразрядно и имеют сложность $O(n)$.

2 Исходный код

Функция или метод	**Описание**
Compare(const TLongNumb &first, TLongNumb &second)	сравнение чисел
PowAssist(TLongNumb &n, TLongNumb &exp)	вспомогательная функция
DltLeadZero()	удаление ведущих нулей
ColumnMult(TLongNumb &first, TLongNumb &second)	умножение столбиком
SmallMult(TLongNumb &first,TLongNumb &second)	умножение на малое
RightShift()	правый сдвиг
Div(TLongNumb &first, TLongNumb &second)	деление

```cpp
#include <iostream>
#include <cmath>
#include <iomanip>
#include <cstdlib>
#include <algorithm>
#include <exception>
#include <vector>

class TLongNumb {
    std::vector<unsigned long long> longNumber;
    const unsigned long long base = 1000*1000*1000;

public:
    TLongNumb() = default;
    TLongNumb(const std::string &str);
    TLongNumb(const TLongNumb &ln);
    TLongNumb &operator=(const TLongNumb &second);
    bool operator<(const TLongNumb &second);
    bool operator>(const TLongNumb &second);
    bool operator==(const TLongNumb &second);
    bool operator<=(const TLongNumb &second);
    bool operator>=(const TLongNumb &second);

    friend TLongNumb Pow(TLongNumb &n, const TLongNumb &exp);
    friend TLongNumb operator+(const TLongNumb &first, const TLongNumb &second);
    friend TLongNumb operator-(const TLongNumb &first, const TLongNumb &second);
    friend TLongNumb operator*(const TLongNumb &first, const TLongNumb &second);
    friend TLongNumb operator/(const TLongNumb &first, const TLongNumb &second);
    friend std::ostream &operator<<(std::ostream &out, const TLongNumb &num);

private:
    friend int Compare(const TLongNumb &first, const TLongNumb &second);
    friend TLongNumb PowAssist(TLongNumb &n, const TLongNumb &exp);
    void DltLeadZero();
    friend TLongNumb ColumnMult(const TLongNumb &first, const TLongNumb &second);
    friend TLongNumb SmallMult(const TLongNumb &first, const TLongNumb &second);
    void RightShift();
    friend TLongNumb Div(const TLongNumb &first, const TLongNumb &second);
};
```

3 Консоль

```
zebr@DESKTOP-T2MUUB0:/mnt/d/Study/DA_lab6/cmake-build-debug$ ./DA_lab6
38943432983521435346436
354353254328383
+
90409438473849324729384738 43
2343543
-
972323
2173937
>
2
3
-
3894343333787468967 4819
90409438473849324729361303 00
false
Error
```

4 Выводы

Выполнив шестую лабораторную работу по курсу «Дискретный анализ», я научился реализовывать различные операции с длинной арифметикой. Кроме того стоит отметить что данное задание оказалось во многих моментах очень интуитивным, так как многие операции выполнялись с помощью алгоритмов, которые похожи на дейтвия с числами, делающиеся вручную. Длинные числа применяются во многих областях науки и техники, надеюсь знания полученные в ходе этой лабораторной пригодятся мне в будущем.

Список литературы

[1] *Длиииииииинная арифметика - Хабр*
URL: `https://habr.com/ru/post/172285/`
(дата обращения: 30.02.2021).

[2] *StackOverflow*
URL `https://stackoverflow.com/questions/45649540/45649702`
(дата обращения: 30.02.2021).

[3] *github*
URL `https://gist.github.com/ar-pa/957297fb3f88996ead11`
(дата обращения: 30.02.2021).

[4] *github*
URL `https://github.com/toshunster/MAI-da/tree/master/lab6`
(дата обращения: 30.02.2021).