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: Т. А. Габдуллин
Преподаватель: А. А. Кухтичев
Группа: М8О-206Б
Оценка:
Подпись:

Москва, 2019

Лабораторная работа №6

Задача: Необходимо разработать программную библиотеку на языке 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 (*).
- Деление (/).
- Возведение в степень ($\hat{}$).

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**.

Ограничения.

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.

Входный файл состоит из последовательностей заданий, каждое задание состоит их трех строк:

1. Первый операнд операции.
2. Второй операнд операции.

3. Символ арифметической операции или проверки условия (+, -, *, ; /, >, <, =).

Числа, поступающие на вход программе, могут иметь «ведущие нули».

Формат выходных данных

Для каждого задания из входного файла нужно распечатать результат на отдельной строке в выходном файле:

1. Числовой результат для арифметических операций.
2. Строку Error в случае возникновения ошибки при выполнении арифметичесокй операции.
3. Строку true или false при выполнении проверки условия.

В выходных данных вывод чисел должен быть нормализован, то есть не содержать в себе «ведущих» нулей.

1 Метод решения

Длинная арифметика — это метод работы с огромными числами, которые не влезают ни в один стандартный числовой тип данных.

Основная идея заключается в том, что число хранится в виде массива его цифр. Цифры могут храниться в той или иной системе счисления. Если использовать десятичную степень - мы будем хранить по 1 цифре в каждом элементе массива, что является не самым эффективным способом, поэтому стоит воспользоваться в б´льшими степенми.

Я использовал беззнаковый тип *std:size_ t*, который имеет предел до $2 * 2^{31} - 1$.

Числа могут иметь любую систему счисления, которую укажет пользователь, поэтому в каждой операции проверяется совместимость двух чисел. Если они несовместимы, то программа завершается с указанием ошибки.

После каждой операции из результата удаляются все лидирующие нули.

1 Проверка условия.

Сравниваем длины - если есть различия, выдаем их. В противном случае идем от меньшего разряда к большему, проверяя на различия. Если находим разницу - выдаем ее.

Линейная оценка: $\emptyset(min\{n, m\})$, где m,n — количество разрядов двух чисел.

2 Сложение.

Идем от самого маленького разряда, до самого большого, при этом длина прохода будет $max\{n, m\}$. Складываем каждые разряды методом сложения в столбик, перенося единицу, в случае необходимости, в старший разряд.

Линейная оценка: $\emptyset(max\{n, m\})$, где m,n — количество разрядов двух чисел.

3 Вычитание.

Сперва проверим, чтобы первый операнд был больше либо равен второго, чтобы не получить отрицательный результат. Операция происходит аналогично сложению, только при отрицательности сложения двух разрядов берется разряд из следующего разряда первого операнда.

4 Умножение.

Как и в обычном умножении столбиком идем по разрядам от младшего к старшему и производим умножение разрядов, перенося лишние десятки в старший разряд.

Линейная оценка: $\emptyset(n * m)$, где m,n — количество разрядов двух чисел.

5 Возведение в степень.

Данную операцию можно решить перемножением исходного числа «в лоб», а можно воспользоваться бинарным возведением в степень

Линейная оценка: $\emptyset(n * \log d)$, где n — количество разрядов числа, а d — степень числа.

6 Деление.

На каждом шагу имеем значение делимого, которое мы пытаемся уменьшить на максимальное количество раз делимым, это максимальное количество раз будем находить **бинарным поиском**.

Линейная оценка: $\emptyset(n * \log m)$, где m,n — количество разрядов двух чисел.

2 Листинг

```
#include <algorithm>
#include <iostream>
#include <iomanip>
#include <vector>
#include <string>
using namespace std;
class TLongNumber {
public:
  TLongNumber(istream &is, int base);
  TLongNumber(string num, int base);
  TLongNumber(vector <int> num, int base);
  TLongNumber(int size, int base);

   TLongNumber(){};
  ~TLongNumber(){};

  int Size () const { return number.size(); }
  bool Divisibility(int a) const;

  friend ostream& operator<<(ostream& os, const TLongNumber &a);

  bool operator ==(const TLongNumber &x) const;
  bool operator > (const TLongNumber &x) const;
  bool operator < (const TLongNumber &x) const;

  TLongNumber operator + (const TLongNumber &a) const;
  TLongNumber operator - (const TLongNumber &a) const;
  TLongNumber operator * (const TLongNumber &a) const;
  TLongNumber operator / (const TLongNumber &a) const;
  TLongNumber operator ^ (const TLongNumber &a) const;

  TLongNumber operator * (const int &a) const;



private:
  void ShiftUp ();
  void ZeroSplit();
  int base;
  vector<int> number;


};
```

3 Выводы

Благодаря данной лабораторной работе я укрепил свои знания в области длинных чисел, познакомился с такими алгоритмами, как бинарное возведение в степень, быстро умножение алгоритмом Карацубы, который дает хороший временной показатель для длинных чисел, но работает дольше для маленьких чисел.

Список литературы

[1] Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ, 2-е издание. — Издательский дом «Вильямс», 2007. Перевод с английского: И. В. Красиков, Н. А. Орехова, В. Н. Романов. — 1296 с. (ISBN 5-8459-0857-4 (рус.))

[2] Дональд Кнут, «Искусство программирования», том 2, «Получисленные алгоритмы», 3-е издание. Глава 4.3, «Арифметика многократной точности», стр. 304–335.

[3] *Алгоритм деления двух чисел*
URL:`https://mindhalls.ru/big-number-in-c-cpp-add-sub/`

[4] *Умножение чисел методом Карацубы*
URL:`https://habrahabr.ru/post/124258/`

[5] *Перегрузка операторов*
URL:`https://habrahabr.ru/post/132014/`.