

Теоретический материал

Рассматриваемые алгоритмы применимы, прежде всего, к длинным и очень длинным целым числам (состоящими, например из 100 или 1000 цифр). Такие числа не могут быть записаны в переменную стандартных типов (int, long и т.п.). Поэтому программная реализация арифметических алгоритмов будет предполагать работу с массивом, каждый элемент которого будет представлять собой цифру в соответствующем разряде длинного числа. Массив можно создавать статически или динамически. Предпочтительный вариант – динамическое создание массивов (на языке C++ с помощью указателей). При этом в отдельной переменной необходимо хранить размер массива (количество цифр в длинном числе).

Пример 1 демонстрирует создание динамического массива для числа, содержащего задаваемое с клавиатуры количество цифр. Массив заполняется случайными цифрами от 0 до 9 (старший разряд не может быть нулевым в этом случае). При этом в нулевом элементе массива содержится младший разряд, а в последнем – старший. То есть на печать массив нужно выводить наоборот, чтобы число отображалось правильно.

Пример 2 демонстрирует сравнение двух больших чисел. Это необходимо, например, при реализации операции вычитания для двух больших чисел. Генерация большого числа реализована через функцию

Сложение и вычитание

Базовые алгоритмы сложения, вычитания, умножения и деления чисел – хорошо известные алгоритмы выполнения действия «в столбик». Вместе с тем вся машинная арифметика сводится к одной операции сложения двух двоичных чисел. В этой связи, отдельного рассмотрения заслуживают задачи ускорения операции сложения. Основная идея ускорения состоит в раздельном рассмотрении поразрядного сложения и переносов. В двоичной системе это реализуется следующим образом, см. рисунок ниже, пример $0100011100 + 0101110111 = 1010010011$.

0	1	0	0	0	1	1	1	0	0
0	1	0	1	1	1	0	1	1	1
	0	0	1	1	0	1	0	1	1
1	0	0	0	1	0	1	0	0	0
1	0	0	1	0	0	0	0	1	1
0	0	0	1	0	1	0	0	0	0
1	0	0	0	0	1	0	0	1	1
0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0

Рисунок 1 – Способ ускорения операции сложения

Здесь в качестве промежуточных значений используются два числа, промежуточная сумма и число переносов. Процесс завершается тогда, когда число переносов полностью обнуляется. На первом шаге суммируются исходные числа, на каждом последующем – промежуточная сумма и число переносов предыдущего шага. Промежуточная сумма формируется согласно следующему правилу: *k-ый разряд суммы равен 0, если равна 0 или 2 сумма k-ых разрядов слагаемых, k-ый разряд суммы равен 1, если равна 1 сумма k-ых разрядов слагаемых.*

Число переносов формируется согласно следующему правилу: *самый младший разряд числа всегда равен 0. k-ый разряд равен 1, если сумма (k-1)-ых разрядов равна 2. В иных случаях разряд числа переносов равен 0.*

Умножение

Умножение столбиком можно назвать естественным алгоритмом, в том смысле, что столбиковое умножение непосредственно опирается на таблицу умножения цифр и определение операции умножения через сложение. Естественные идеи такого рода хороши для реализации, так как предполагают только некоторые технические сложности и не более того, но вряд ли так можно получить максимум эффективности.

Метод Карацубы

Карацуба высказал достаточно простую идею, позволяющую умножать числа существенно быстрее. Его идея основана на следующем очевидном соотношении:

$$4ab = (a+b)^2 - (a-b)^2, \text{ откуда } ab = \frac{(a+b)^2 - (a-b)^2}{4}$$

Основная идея – разбить исходное число на два меньших, это должно дать экономию. Идея действительно выигрышная. Число X можно разными способами представить в виде суммы двух: $X = X_1 + X_2$. Найдем такое представление, что длина X_1 равна длине X и половина младших разрядов X_1 равна нулю. Тогда длина X_2 равна половине длины X . Пример:

$$4578394\ 9002338 = 45783940000000 + 9002338 = 4578394 * 10^7 + 9002338.$$

При этом если числа являются разной длины или имеют нечетную длину, то целесообразно добавить старший «нулевой разряд». Например, если даны числа 37656 и 6567863, то их следует рассматривать в таком виде:

$$00037656 = 0003 * 10^4 + 7656;$$

$$06567863 = 0656 * 10^4 + 7863;$$

Будем рассматривать произведение двух чисел в виде $(ax+b)(cx+d)$, где $x=10^k$ (в примере выше $k=4$). Имеет место следующая цепочка равенств

$$\begin{aligned}(ax + b)(cx + d) &= acx^2 + (ad + bc)x + bd = \\ &= acx^2 + ((a + b)(c + d) - ac - bd)x + bd.\end{aligned}$$

Для наших двух чисел имеем:

$$(0003 * 10^4 + 7656) * (0656 * 10^4 + 7863) =$$

$$0003 * 0656 * 10^8 + ((0003 + 7656) * (0656 + 7863) - 0003 * 0656 - 7656 * 7863) * 10^4 + 7656 * 7863$$

В этом выражении произведения $0003 * 0656$ и $7656 * 7863$ повторяются дважды, а всего нужно посчитать 3 произведения четырехразрядных чисел:

- 1) $0003 * 0656$
- 2) $7656 * 7863$
- 3) $(0003 + 7656) * (0656 + 7863) = 7659 * 8519$

Каждое произведение четырехразрядных чисел можно разбить на произведения двухразрядных, а произведения двухразрядных – на произведения одnorазрядных чисел.

Пример для умножения четырехразрядных чисел методом Карацубы:

Вычислим $1213 \cdot 2311$:

- заметим, что $12 + 13 = 25$, $23 + 11 = 34$ и вычислим $25 \cdot 34$:
 - заметим, что $2 + 5 = 7$, $3 + 4 = 7$ и вычислим $7 \cdot 7 = 49$
 - вычислим $2 \cdot 3 = 6$
 - вычислим $5 \cdot 4 = 20$
 - складывая результаты, получим, что $25 \cdot 34 = 6 \cdot 10^2 + (49 - 6 - 20) \cdot 10 + 20 = 850$
- вычислим $12 \cdot 23$ как $12 \cdot 23$:
 - заметим, что $1 + 2 = 3$, $2 + 3 = 5$ и вычислим $3 \cdot 5 = 15$
 - вычислим $1 \cdot 2 = 2$
 - вычислим $2 \cdot 3 = 6$
 - складывая результаты, получим, что $12 \cdot 23 = 2 \cdot 10^2 + (15 - 2 - 6) \cdot 10 + 6 = 276$
- вычислим $13 \cdot 11$ как $13 \cdot 11$:
 - заметим, что $1 + 3 = 4$, $1 + 1 = 2$ и вычислим $4 \cdot 2 = 8$
 - вычислим $1 \cdot 1 = 1$
 - вычислим $3 \cdot 1 = 3$
 - складывая результаты, получим, что $13 \cdot 11 = 1 \cdot 10^2 + (8 - 1 - 3) \cdot 10 + 3 = 143$
- складывая результаты, получим, что $1213 \cdot 2311 = 276 \cdot 10^4 + (850 - 276 - 143) \cdot 10^2 + 143 = 2803243$

Деление. Алгоритм с использованием метода половинного деления

Пусть некоторое время нас интересует только частное. Частное можно представить как неизвестную величину в уравнении вида:

$$A - Bp = 0, \text{ где } A - \text{ делимое, } B - \text{ делитель, } p - \text{ частное.}$$

Как известно, такие уравнения можно решать методом половинного деления. Здесь $f = A - Bp$, левая граница отрезка, содержащего корень, есть 0 и правая граница равна делимому A . Решением уравнения будет частное p . Далее остаток находится просто. Если частное получено с точностью до целого, то остаток легко определяется из формулы $A = Bp + q \Leftrightarrow q = A - Bp$

Вместо метода половинного деления можно использовать и другие методы: метод хорд или касательных

Пример 1

Задача:

Написать на языке C++ программу, в которой создается динамический массив для хранения длинного целого числа (более 10 цифр). Размер массива задается с клавиатуры. Цифры генерируются случайным образом.

Решение:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      setlocale(LC_ALL, "Russian");
8      srand((unsigned)time(NULL)); //Для генератора случайных чисел
9      srand((unsigned)rand()); //Для генератора случайных чисел
10     int count = 0; //Переменная для хранения размера числа
11     cout << "Введите размер числа (количество цифр): ";
12     cin >> count;
13
14     int* num = new int[count]; //Массив для длинного числа
15     for (int i = 0; i < count - 1; i++)
16     {
17         num[i] = rand() % 10; //заполнение цифр кроме старшей
18     }
19
20     //Генерация старшей ненулевой цифры
21     int hnum = 0;
22     while (true)
23     {
24         hnum = rand() % 10;
25         if (hnum != 0) break;
26     }
27     num[count - 1] = hnum;
28
29     // Вывод числа
30     cout << "Ваше число:" << endl;
31     for (int i = count - 1; i >= 0; i--)
32     {
33         cout << num[i] << " ";
34     }
35     cout << endl;
36
37     delete[] num;
38 }
```

Ответ:

Консоль отладки Microsoft Visual Studio

Введите размер числа (количество цифр): 50

Ваше число:

8 3 1 1 4 3 3 2 1 3 4 5 9 7 2 9 8 1 6 7 5 8 6 1 2 2 3 2 4 1 6 7 2 5 1 1 7 1 5 5 4 2 8 8 4 9 0 4 7

Пример 2

Задача:

Написать на языке C++ программу для сравнения двух больших чисел (более 10 цифр)

Решение:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int* big_num(int count)
6  {
7      int* num = new int[count]; //Массив для длинного числа
8      for (int i = 0; i < count - 1; i++) num[i] = rand() % 10; //заполнение цифр кроме старшей
9
10     //Генерация старшей ненулевой цифры
11     int hnum = 0;
12     while (true)
13     {
14         hnum = rand() % 10;
15         if (hnum != 0) break;
16     }
17     num[count - 1] = hnum;
18     return num;
19 }
20
21 int main()
22 {
23     setlocale(LC_ALL, "Russian");
24     int count = 0; //Переменная для хранения размера числа
25     cout << "Введите размер числа (количество цифр): ";
26     cin >> count;
27
28     //Первое число
29     int* num1 = big_num(count);
30     cout << "Первое число: ";
31     for (int i = count - 1; i >= 0; i--) cout << num1[i];
32     cout << endl;
33
34     //Второе число
35     int* num2 = big_num(count);
36     cout << "Второе число: ";
37     for (int i = count - 1; i >= 0; i--) cout << num2[i];
38     cout << endl;
39
40     //Сравнение чисел
41     for (int i = count - 1; i >= 0; i--)
42     {
43         if (num1[i] < num2[i])
44         {
45             cout << "Первое число меньше второго" << endl; break;
46         }
47         else if (num1[i] > num2[i])
48         {
49             cout << "Первое число больше второго" << endl; break;
50         }
51         else if (i == 0)
52         {
53             cout << "Числа равны" << endl;
54         }
55     }
56     delete[] num1; delete[] num2;
57 }
58
```

Ответ:

Консоль отладки Microsoft Visual Studio

Введите размер числа (количество цифр): 77

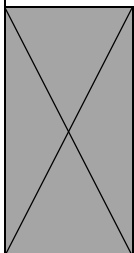
Первое число: 6009056842068205689139772478311433213459729816758612232416725117155428849047

Второе число: 3662818995337777170641002145756106273790245163670489531928873694816606443983

Первое число больше второго

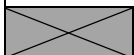
Задание 1

Задача:

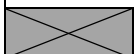


Написать программу для сложения и вычитания двух больших чисел в столбик. Для представления чисел использовать массив, в котором будут храниться цифры числа. При реализации процедуры вычитания предпочтительно из большего вычитать меньшее. Для этого первоначально определять, какое из двух чисел больше

Решение:

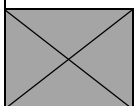


Ответ:



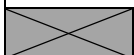
Задание 2*

Задача:



Написать программу для реализации ускоренного сложения чисел в двоичной системе (аналогично примеру на рис. 1)

Решение:

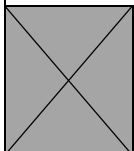


Ответ:



Задание 3*

Задача:



В задании 2 была реализована идея ускорения суммирования двоичных чисел. Реализуйте сложение десятичных чисел с использованием подобного механизма ускорения.

Решение:

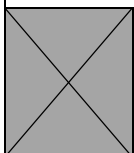


Ответ:



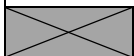
Задание 4

Задача:

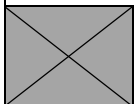


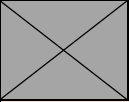
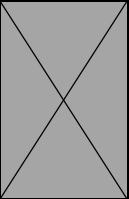
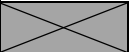
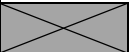
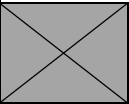
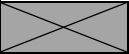
Написать программу для умножения двух больших чисел столбиком. Обратите внимание на количество цифр в произведении: их скорее всего будет больше, чем в исходных числах

Решение:



Ответ:



	
Задание 5*	
Задача:	
	Написать программу для умножения двух больших чисел методом Карацубы. Код программы объединить с кодом для задания 4, чтобы иметь возможность сравнить результат умножения одних и тех же чисел столбиком и методом Карацубы
Решение:	
	
Ответ:	
	
Задание 6	
Задача:	
	Написать программу для нахождения частного и остатка от деления двух чисел методом половинного деления
Решение:	
	
Ответ:	
