Информатика (основы программирования)

Лекция 2

Алгоритмы обработки целых чисел

Автор: Бабалова И.Ф.
Доцент, каф.12

2023 г.

Запись числа в памяти компьютера

- Операции над целыми числами не могут уменьшить точность исходных данных, а операции над вещественными числами могут существенно ухудшить точность результатов. Поэтому вопросам улучшения точности решения задач и соответствующего уменьшения времени выполнения операций посвящены многие разделы теории алгоритмов.
- Давайте рассмотрим интересный алгоритм для уменьшения количества умножений. Каждое уменьшение количества операций для вещественных чисел позволяет повысить точность решения задачи

Пример улучшения алгоритма умножения

- Алгоритм умножения (возведение в степень любого числа), называемый в математике «индийским алгоритмом», сформулирован так :
  - Для вычисления натуральной n-й ($n > 0$) степени целого или вещественного числа x рассмотрим следующие формулы:

    при n=1 $x^n = x$;

    при n>1 $x^n = x^{n \, mod 2} \left(x^{n \, div \, 2}\right)^2 \quad \rightarrow \quad x^5 = x \, * (x^2)^2$

- Сокращение количества умножений обеспечивается за счёт того, что при каждом выполнении вызова операции происходит не более одного деления, возведения в квадрат и умножения, поэтому общее количество арифметических операций не больше $3 \log_2 n$. При больших значениях n это число будет существенно меньше обычных n - 1 умножений. Например, при n = 1000 это примерно 30 операций

Ряд Фибоначчи (1)

- Ряд Фибоначчи используется физиками, архитекторами, строителями очень часто, так как отношение двух соседних чисел ряда стремится к значению золотого сечения – числу $1.6180339\ldots$. В правильном изображении пятиконечной звезды каждый отрезок делится пересекающим его отрезком в отношении, равном значению золотого сечения.
  - Элементы ряда Фибоначчи считаются по формуле $F_{n+1} = F_n + F_{n-1}$
  - Получаем последовательность чисел :

    $1, 1, 2, 3, 5, 8, 13$ и т.д.

    Отношение $8/5$ уже равно 1.6.
- При увеличении количества членов ряда повышается точность вычисления золотого сечения. Если нужны два числа Фибоначчи для достаточно больших N, задача может быть решена обычным циклом из N сложений.
- Значительно улучшить сходимость алгоритма можно, воспользовавшись линейной алгеброй.

Ряд Фибоначчи (2)

- Рассмотрим вектор F и матрицу V:
- $F = \begin{vmatrix} 1 \\ 1 \end{vmatrix} \quad V = \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}$ → Умножив матрицу на вектор, получим следующее число Фибоначчи.
- Если вектор F равен (1; 1), то умножив матрицу V на вектор F, получим новый вектор
- $\begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} 1 \\ 1 \end{vmatrix} = \begin{vmatrix} 0 * 1 + 1 * 1 \\ 1 * 1 + 1 * 1 \end{vmatrix} = \begin{vmatrix} 1 \\ 2 \end{vmatrix}$-------- Следующее значение числа Фибоначчи
- Таким образом, делаем вывод, что не надо выполнять n сложений для вычисления n-ого числа Фибоначчи, а можно возвести матрицу V в n-ю степень. Это чисто математическое исследование существенно уменьшает количество вычислений, и сложность алгоритма уменьшается практически в n раз.

Запись числа в памяти компьютера (1)

- Целые числа

Данные типа int занимают одно машинное слово – 4 байта – 32 разряда (бита):

+6 0 000 0000 0000 0000 0000 0000 0000 0110 (прямой код)

-6 1 111 1111 1111 1111 1111 1111 1111 1010 (дополнительный код)

Стандартная длина – 4 байта (1 байт = 8 бит)

Запись числа в памяти компьютера (2)

- Вещественные числа
  - Размер числа в зависимости от типа компьютера 4, 8, 12 байтов

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	0	1	1	1	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

1 байт + 1 бит знака числа, содержащие знак числа и порядок числа

Знак порядка не нужен, так как увеличивается на 2^8-1 и будет всегда положительным

Дано число **2021,125 – 11111100101,00**

Число в памяти записывается в нормализованной форме

2,021125*10³ – ,111110010100*¹⁰¹⁰

(в двоичном виде) Не записывается в памяти и 1 в нормализованной мантиссе

Алгоритмизация операций с вещественными числами

- Формула для записи вещественного числа:

$$x = \sum_{i=m-n+1}^{m} a_i \cdot 10^i ,$$

где m – порядок числа (степень основания системы счисления у старшей цифры числа);
a_i – цифры числа в используемой системы счисления
n – количество цифр в записи числа

- Выполнение операций с вещественными числами всегда приводит к увеличению погрешности из-за ограниченности формата записи чисел в компьютере. Определено, что за 10 операций сложения теряется один разряд в точности записи вещественного числа.

Оценка точности записи результата вычислений

- **Определение**:
  - В записи числа n цифр верные, если абсолютная погрешность записи числа не превышает половины единицы разряда n –ой значащей цифры:

$$\varepsilon \le \frac{1}{2} \cdot 10^{m-n+1}$$

234,1531 – число с верными цифрами

Погрешность его равна:

$\Delta = 0{,}5 \cdot 10^{-4}$

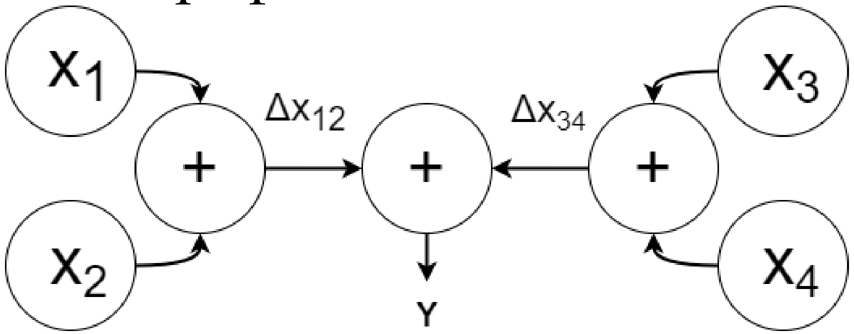
$m = 2,\ n = 7$

$m - n + 1 = 4$

Вычисления с заданной точностью

- Сложение вещественных чисел
  - $y = x_1 + x_2 + x_3 + x_4$
- В записи каждого вещественного числа есть ошибка. При выполнении любой арифметической операции ошибки только увеличиваются, так как количество разрядов для записи числа постоянно.
- Рекомендации. **Сложение последовательности чисел начинается с наименьших чисел.**
- Для выполнения простейшей операции сложения для достижения высокой точности результата исходные данные упорядочиваются

1. Граф сложения 4-х чисел:



$\Delta Y = \Delta x_{12} + \Delta x_{34}$

2. Сложении чисел в порядке записи чисел:

ΔY
$= \Delta x_1 + \Delta x_2 + (\Delta x_{12} + \Delta x_3) + (\Delta x_{123})$
$+ \Delta x_4$

Ошибка выполнения операции накапливается

Спасибо за внимание