Дисциплина

«РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ»

Лекция 12

РЕКУРСИВНЫЕ ФУНКЦИИ

Лектор: Михаил Васильевич Ульянов,

muljanov@mail.ru, **8 916 589 94 04**

1. Три ипостаси термина рекурсивная функция

Выделим три области, в которых термин «рекурсивная функция» используется с различной семантической нагрузкой:

- **Теория алгоритмов**. Аппарат частично рекурсивных функций (Чёрч и Клини) является одним их четырех основных формализмов теории алгоритмов.

  Подробнее см., например, http://mit.spbau.ru/sewiki/images/7/7d/Mlc14.pdf
- **Теория рекурсии в математике** — предмет данной лекции
- **Программирование**. Применение рекурсии в программировании — предмет второй части нашего лекционного курса. Изложение опирается на книгу:

Головешкин В.А., Ульянов М.В. Теория рекурсии для программистов. — М.: ФИЗМАТЛИТ, 2006. — 296 с.

2. Способы задания функции

Определение А.Н. Колмогорова — **функция** *из множества A в множество B это отображение, униморфное по второй координате*.

Способы задания:

1. Табличный способ — перечисление значений аргумента и соответствующих значений функции. Способ, который мы используем всегда при работе с экспериментальными данными.
2. Способ на основе модели функций. Вводится понятие элементарной функции (аналогично элементарным операциям в модели вычислений) и операции, позволяющие строить более сложные конструкции из элементарных. Пример $e^{\sin x + \ln(\cos x)^2}$.
3. Рекурсивный способ — определение функции через ее начальные и предыдущие значения.

3. Рекурсивное задание функции

! Мы будем далее рассматривать целочисленные функции целочисленного аргумента — $f(n): Z^+ \to Z^+$. Задание функции происходит в два этапа:

Первый этап. Функция $f(n)$ задается непосредственно в виде числовых значений для конечного множества начальных значений аргумента

$$n: 0 \le n \le m.$$

Второй этап. Задается метод или формула, которые позволяют, зная все значения функции $f(n)$ при $n \le k$ $(k > m)$, вычислить ее значения при $n = k + 1$, то есть найти $f(k+1)$ — мы получаем **рекуррентные соотношения**, описывающие рекурсивно заданную функцию $f(n)$. Полученная запись этих двух этапов, обычно объединяется фигурной скобкой

$$\begin{cases} f(1) = f_1, f(2) = f_2, \dots, f(m) = f_m; \\ f(n+1) = g(f_n, f_{n-1}, \dots, f_1, n, const), n \ge m. \end{cases}$$

4. Рекуррентное соотношение

Такая форма записи, объединяющая два этапа задания функции

$$\begin{cases} f(1) = f_1, f(2) = f_2, \dots, f(m) = f_m; \\ f(n+1) = g(f_n, f_{n-1}, \dots, f_1, n, const), \quad n \geq m. \end{cases}$$

носит название **рекуррентного соотношения**. Таким образом, мы будем говорить, что рекуррентное соотношение определяет рекурсивно заданную функцию $f(n)$. Чем занимается математическая теория рекурсии? Если мы имеем некоторое рекуррентное соотношение, то можем ли мы представить эту функцию в явном виде? Иными словами, можем ли мы выразить $f(n)$ через элементарные функции. Это называется решить рекуррентное соотношение. Теория рекурсии разрабатывает методы решения рекуррентных соотношений и соответствующую классификацию, которая опирается на возможность применения тех или иных методов их решения к рекуррентным соотношениям в данном классе.

5. Примеры рекуррентных соотношений

Давайте попробуем решить некоторые рекуррентные соотношения:

$$1.\ \begin{cases} f(1) = 1; \\ f(n+1) = f(n) + 1, \quad n \geq 1. \end{cases}$$

Ответ очевиден $f(n) = n$. Обратите внимание что, по сути, мы работаем здесь с конечными разностями!

$$2.\ \begin{cases} f(1) = 1; \\ f(n+1) = f(n) + 2n + 1, \quad n \geq 1. \end{cases}$$

Вычислим последовательно значения: $f(2) = f(1) + 2 + 1 = 4$, $f(3) = f(2) + 4 + 1 = 9$.

Мы получаем $f(n) = n^2$, и непосредственно подтверждаем, что последовательная сумма нечетных чисел есть полный квадрат.

$$3.\ \begin{cases} f(1) = 1, f(2) = 1 \\ f(n) = f(n-1) + f(n-2), \quad n \geq 3. \end{cases}$$

Это линейное рекуррентное соотношение второго порядка. Что это за функция? Это числа Фибоначчи и для них есть аналитическое решение (см. книгу). Функция $f(n) \approx \phi^n, \phi \approx 1{,}618$. Аналитическое решение получено Бине в XIX веке.

$$4.\ \begin{cases} f(0) = 1 \\ f(n) = 2f(n-1) \quad n \geq 1. \end{cases}$$

Очевидно, что $f(n) = 2^n$. Обратите внимание на то, что очень важную роль играют начальные условия (задача Коши). Пусть $f(0) = 0$, что такое теперь $f(n)$? А если $f(0) = 7$?

Обратите внимание, что в последних двух примерах мы выражали $f(n)$, а не $f(n+1)$ через предыдущие значения! Это вопросы удобства представления.

$$5.\ \begin{cases} f(0) = 1 \\ f(n) = f(n-1) + f(n-2) + \cdots f(0) + 1 \quad n \geq 1. \end{cases}$$

Эта конструкция называется рекуррентное соотношение с полной предысторией! Что это за функция? Мы можем решить это рекуррентное соотношение? Оказывается, что опять $f(n) = 2^n$.

$$6.\ \begin{cases} f(0) = 1 \\ f(n) = n \cdot f(n-1) \quad n \geq 1. \end{cases}$$

А это рекуррентное соотношение для факториала!

Почему 0!=1? Можем ли мы предложить функцию, распространяющую факториал на все действительные числа? Очень красивое решение принадлежит Л. Эйлеру (L.Euler).

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt,$$

причем $\Gamma(s+1) = s \cdot \Gamma(s)$, $\Gamma(n+1) = n!\ \Gamma(1) = 0! = 1$.

Рассмотрим еще три достаточно полезных примера:

$$7.\ \begin{cases} f(1) = 1; \\ f(n+1) = 0{,}3 \cdot f(n) + 2, \quad n \geq 1, \end{cases}$$

Это соотношение позволяет нам с нужной степенью точности вычислить корень уравнения $0{,}7 \cdot x = 2$, не используя операцию деления. См. книгу!

$$8.\ \begin{cases} f(1) = 1; \\ f(n+1) = \frac{1}{2}\left(f(n) + \frac{a}{f(n)}\right), \quad n \geq 1, \end{cases}$$

Это рекуррентное соотношение позволяет нам получить с нужной степенью точности значение $x = \sqrt{a}$ при положительных значениях числа a. Это один из способов реализации стандартной функции вычисления квадратного корня!

А теперь пример функции двух переменных.

Рассмотрим функцию двух переменных $C(n;k)$, область определения которой: n — целые неотрицательные числа, а k — целые числа. Положим, что

$$C(n;k) = C_n^k = \frac{n!}{k!(n-k)!},$$

при $0 \le k \le n$, и $C(n;k) = 0$ при $k < 0$ и $k > n$. Заданную таким образом функцию можно легко определить как функцию, рекурсивно заданную следующим рекуррентным соотношением

$$9.\ \begin{cases} C(0;k) = 0, \quad k \ne 0, \quad C(0;0) = 1; \\ C(n+1;\ k) = C(n;\ k) + C(n;\ k-1), n \ge 0. \end{cases}$$

И рассмотрим пример функции, для которой нет аналитического решения, но рекуррентное соотношение строится достаточно просто. Нас интересует функция, значение которой есть число единиц в двоичном представлении числа n. Эта функция обозначается $\beta_1(n)$. Рассмотрим начальные значения:

n	0	1	10	11	100	101	110	111	1000
$\beta_1(n)$	0	1	1	2	1	2	2	3	1

Попробуем составить рекуррентное соотношение для $\beta_1(n)$. Заметим, что при умножении числа на 2 число единиц не изменяется, а для того, чтобы добавить единицу число необходимо умножить на 2 и прибавить 1.

$$\begin{cases} \beta_1(0) = 0;\ \beta_1(1) = 1; \\ \beta_1(2n) = \beta_1(n) \\ \beta_1(2n+1) = \beta_1(n) + 1,\ n \geq 1 \end{cases}$$

! Если для необходимых нам вычислений мы можем получить рекуррентное соотношение, и если язык программирования позволяет выполнять рекурсивные вызовы, то у нас в руках достаточно интересный аппарат реализации вычислений.