Построение и анализ алгоритмов – Лекция 2. Анализ рекуррентных соотношений. Основная теорема

Олег Смирнов
oleg.smirnov@gmail.com

6 октября 2011 г.

Содержание

Цель лекции

- Строгие математические определения для O-, Ω- Θ-нотации
- Методы решения рекурентностей: подстановка, дерево рекурсии и основная теорема

1 Асимптотическая нотация

1.1 O-обозначения

В случае, когда необходимо определить только *асимптотическую верхнюю границу*, используют O-обозначения:

$$f(n) = O(g(n)) \Rightarrow \exists c, n_0 > 0 \text{ такие, что } 0 \leqslant f(n) \leqslant cg(n), \forall n > n_0$$

$O(g(n))$ можно рассматривать как множество функций:

$$O(g(n)) = \{f(n) : \exists c, n_0 > 0 \text{ такие, что } 0 \leqslant f(n) \leqslant cg(n), \forall n > n_0\}$$

Пример:

$$2n^3 = O(n^3) \text{ для } c = 1, n_0 = 2 \text{ или } 2n^3 \in O(n^3)$$

Определение через множество можно использовать в качестве "макроса" – O-нотация в формуле обозначает некоторую функцию из соответствующего семейства.

Пример:

$$n^2 + O(n) = O(n^2) \text{ означает } \forall f(n) \in O(n) : \exists h(n) \in O(n^2) : n^2 + f(n) = h(n)$$

1.2 Ω-обозначения

Для определения *асимптотической нижней границы* есть Ω-обозначение:

$$\Omega(g(n)) = \{f(n) : \exists c, n_0 > 0 \text{ такие, что } 0 \leqslant cg(n) \leqslant f(n), \forall n > n_0\}$$

Пример:

$$\sqrt(n) = \Omega(\lg n) \text{ для } c = 1, n_0 = 16$$

1.3 Θ-обозначения

Для точной оценки используется Θ-обозначение. Его можно ввести несколькими способами:

- Теорема: для любых двух функций $f(n)$ и $g(n)$ соотношение $f(n) = \Theta(g(n))$ верно тогда и только тогда, когда $f(n) = O(g(n))$ и $f(n) = \Omega(g(n))$
- Пересечение множеств: $f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \cap \Omega(g(n))$

Пример:

$$\frac{1}{2}n^2 - 2n = \Theta(n^2)$$

1.4 o- и ω-обозначения

o- и ω-обозначения являются версиями определений O и Ω, которые выполняются *для любой константы* c, т.е. не являются асимптотически строгими. Формально:

$$o(g(n)) = \{f(n) : \forall c > 0 : \exists n_0 > 0 \text{ такое, что } 0 \leqslant f(n) \leqslant cg(n), \forall n > n_0\}$$

Функция $f(n)$ пренебрежимо мала по сравнению с функцией $g(n)$ при n стремящемся к бесконечности, т.е.:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

Пример:

$$2n^2 = o(n^3) \text{ для } n_0 = \frac{2}{c}$$

Аналогично:

$$\omega(g(n)) = \{f(n) : \forall c > 0 : \exists n_0 > 0 \text{ такое, что } 0 \leqslant cg(n) \leqslant f(n), \forall n > n_0\}$$

Пример:

$$\sqrt(n) = \omega(\lg n) \text{ для } n_0 = 1 + \frac{1}{c}$$

1.5 Некоторые свойства

Из определений вытекают некоторые свойства асимтотических сравнений:

- Транзитивность

$$f(n) = \Theta(g(n)) \wedge g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$$
$$f(n) = O(g(n)) \wedge g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$$
$$f(n) = \Omega(g(n)) \wedge g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$$
$$f(n) = o(g(n)) \wedge g(n) = o(h(n)) \Rightarrow f(n) = o(h(n))$$
$$f(n) = \omega(g(n)) \wedge g(n) = \omega(h(n)) \Rightarrow f(n) = \omega(h(n))$$

- Рефлексивность

$$f(n) = \Theta(f(n))$$
$$f(n) = O(f(n))$$
$$f(n) = \Omega(f(n))$$

- Симметричность

$$f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n))$$

- Перестановочная симметрия

$$f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$$
$$f(n) = o(g(n)) \iff g(n) = \omega(f(n))$$

Таким образом можно провести аналогию между символами нотации и операциями сравнения рациональных чисел:

$$f(n) = \Theta(g(n)) \approx f(n) = g(n)$$
$$f(n) = O(g(n)) \approx f(n) \leqslant g(n)$$
$$f(n) = \Omega(g(n)) \approx f(n) \geqslant g(n)$$
$$f(n) = o(g(n)) \approx f(n) < g(n)$$
$$f(n) = \omega(g(n)) \approx f(n) > g(n)$$

2 Рекуррентные соотношения

По определению, рекуррентное соотношение – это уравнение или неравенство, описывающее функцию с использованием её самой, но только с меньшими аргументами. Обычно рекуррентное соотношение описывается в виде системы граничных условий и формулы для общего случая, например:

$$T(n) = \begin{cases} \Theta(1), \text{ если } n = 1 \\ 2\Theta(n/2) + \Theta(n), \text{ если } n > 1 \end{cases}$$

Для решения таких соотношений используются несколько методов: подстановки, деревьев рекурсии и основная теорема.

3 Метод подстановки

Метод состоит из трех шагов:

- делается догадка о виде решения
- с помощью метода математической индукции доказывается, что решение правильное
- вычисляются константы

3.1 Примеры решения

$$T(n) = 4T(n/2) + n$$

- случаи O и для Ω рассматриваются отдельно
- догадка: $T(n) = O(n^3)$
- пусть $T(k) \leqslant ck^3$ для $k < n$ (гипотеза)
- необходимо доказать методом мат. индукции, что $T(n) \leqslant cn^3$

$$\begin{aligned}
T(n) &= 4T(n/2) + n \\
&\leqslant 4c(n/2)^3 + n \\
&= (c/2)n^3 + n \\
&= cn^3 - ((c/2)n^3 - n) \\
&\leqslant cn^3
\end{aligned}$$

Неравенство выполняется, когда "остаточная часть" $(c/2)n^3 - n$ больше нуля, т.е. для любой константы $c \geqslant 2$ и $n \geqslant 1$.

Доказательство базового случая:

$$\begin{aligned}
T(n) &= \Theta(1), n < n_0 \\
1 &\leqslant n < n_0 \\
\Theta(1) &\leqslant cn^3 \\
&\text{для всех достаточно больших } c
\end{aligned}$$

Доказательство строгой верхней границы:

- необходимо доказать: $T(n) = O(n^2)$

- пусть $T(k) \leqslant ck^2$ для $k < n$ (гипотеза)

$$\begin{aligned} T(n) &= 4T(n/2) + n \\ &\leqslant 4c(n/2)^2 + n \\ &= cn^2 + n \\ &= O(n^2) \end{aligned}$$

– неверно, т.к. необходимо доказать индукционную гипотезу

$$\begin{aligned} &= cn^2 - (-n) \\ &\leqslant cn^2 \end{aligned}$$

– неверно для всех $c > 0$

Необходимо усилить индукционную гипотезу через вычитание члена более низкого порядка:

- пусть $T(k) \leqslant c_1k^2 - c_2k$ для $k < n$

$$\begin{aligned} T(n) &= 4T(n/2) + n \\ &= 4(c_1(n/2)^2 - c_2(n/2)) + n \\ &= c_1n^2 - 2c_2n + n \\ &= c_1n^2 - c_2n - (c_2n - n) \\ &\leqslant c_1n^2 - c_2n \\ &\text{если } c_2 \geqslant 1 \end{aligned}$$

Верно для достаточно больших c_1. Строгая нижняя граница доказывается по аналогии.

4 Метод деревьев рекурсии

- изображается дерево, в узлах которого находится время, требуемое для выполнения данной подзадачи
- значение времени суммируется в пределах уровня, затем – по всем уровням дерева
- подходит для получения "догадки" о виде решения для метода подстановок

4.1 Примеры решения

$$T(n) = T(n/4) + T(n/2) + n^2$$

Для задачи размером n на каждом шаге рекурсивно решается задача размером $n/4$ и $n/2$. Нерекурсивная часть выполняется за n^2 итераций.

После построения дерева (см. рис. 1 - 6) ответом будет сумма геометрической прогрессии $n^2(1 + 5/6 + (5/6)^2 + (5/6)^3 + ...) = \Theta(n^2)$

5 Основная теорема

Основная теорема применятся для решения рекуррентных соотношений вида

$$T(n) = aT(n/b) + f(n)$$

где $a \geqslant 1$, $b > 1$ – константы, а $f(n)$ – асимптотически положительная функция.

Рекуррентное соотношение описывает время работы алгоритма, в котором задача размером n разбивается на a вспомогательных задач, размером n/b каждая, где a и b – положительные константы. Полученные в результате разбиения подзадачи решаются рекурсивным методом, причем время их решения равно (a/b). Время, требуемое для разбиения задачи и объединения результатов, полученных при решении вспомогательных задач, описывается функцией $f(n)$.

При использовании основного метода функция $f(n)$ сравнивается с $n^{\log_b a}$ и рассматриваются три случая. Интуитивно понятно, что асимптотическое поведение решения рекуррентного соотношения определяется большей из двух функций.

1. Если $f(n) = O(n^{\log_b a - \epsilon})$, для некоторой константы $\epsilon > 0$, т.е. $f(n)$ растет полиноминально медленней чем $n^{log_b a}$ в n^ϵ раз.
   Тогда $T(n) = \Theta(n^{\log_b a})$
2. Если $f(n) = \Theta(n^{\log_b a} \lg^k n)$, т.е. $f(n)$ и $n^{\log_b a}$ растут с одинаковой скоростью с точностью до множителя $\lg^k n$, для константы $k \geqslant 0$.
   Тогда $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$
3. Если $f(n) = \Omega(n^{\log_b a + \epsilon})$, для некоторой константы $\epsilon > 0$, т.е. $f(n)$ растет полиноминально быстрей чем $n^{log_b a}$ в n^ϵ раз
   и $f(n)$ удовлетворяет неравенству $af(n/b) \leqslant cf(n)$ для некоторого $c < 1$
   Тогда $T(n) = \Theta(f(n))$

Важно понимать, что этими тремя случаями не исчерпываются все возможности поведения функции $f(n)$. Между случаями 1 и 2 есть промежуток, в котором функция $f(n)$ меньше функции $n^{\log_b a}$, но не полиноминально меньше. Аналогичный промежуток имеется между случаями 2 и 3, когда функция $f(n)$ не полиноминально больше. Если функция $f(n)$ попадает в один из этих промежутков или, если для нее не выполняется условие регулярности из случая 3, основной метод неприменим.

5.1 Примеры решения

$$T(n) = 4T(n/2) + n$$
$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n$$

Случай 1: $f(n) = O(n^{2-\epsilon})$ для $\epsilon = 1$ Ответ: $T(n) = \Theta(n^2)$

$$T(n) = 4T(n/2) + n^2$$

$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^2$$

Случай 2: $f(n) = \Theta(n^2 \lg^0 n)$ Ответ: $T(n) = \Theta(n^2 \lg n)$

$$T(n) = 4T(n/2) + n^3$$

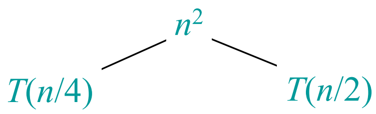
$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = n^3$$

Случай 3: $f(n) = \Omega(n^{2+\epsilon})$, для $\epsilon = 1$ и $4(n/2)^3 \leqslant cn^3$ для $c = 1/2$ Ответ: $T(n) = \Theta(n^3)$

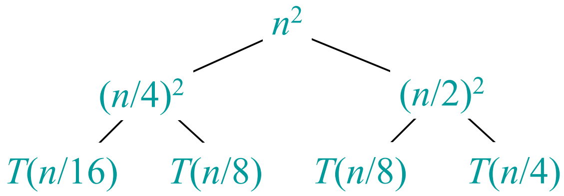
$$T(n) = 4T(n/2) + \frac{n^2}{\lg n}$$

$$a = 4, b = 2 \Rightarrow n^{\log_b a} = n^2; f(n) = \frac{n^2}{\lg n}$$

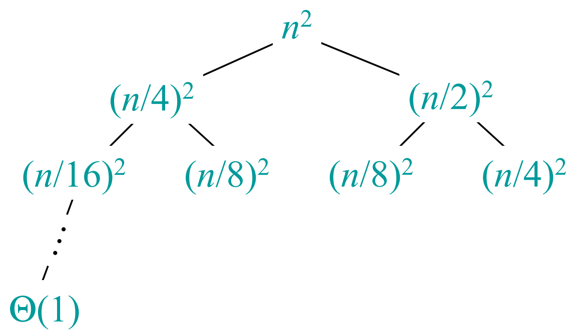
Основной метод неприменим, т.к. для любой константы $\epsilon > 0$ справедливо $n^\epsilon = \omega(\lg n)$



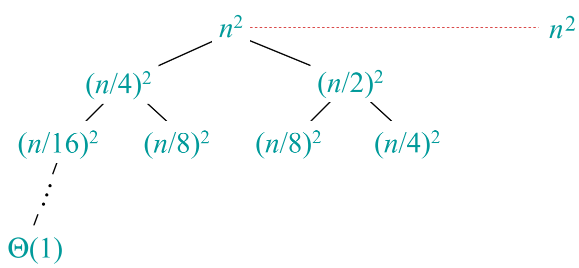
Дерево рекурсии – шаг 1



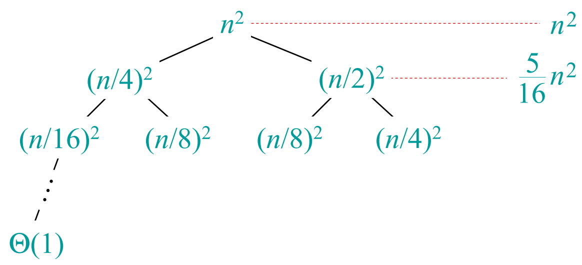
Дерево рекурсии – шаг 2



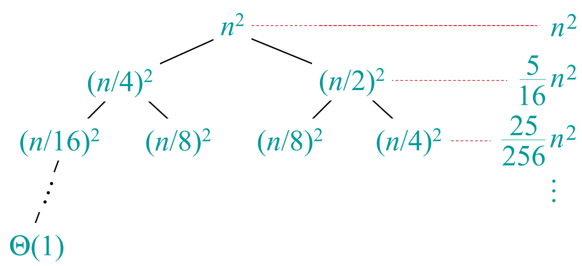
Дерево рекурсии – шаг 3



Дерево рекурсии – шаг 4



Дерево рекурсии – шаг 5



Дерево рекурсии – шаг 6