ПРИМИНЕНИЕ АСИМПТОТИЧЕСКИХ ОБОЗНАЧЕНИЙ ДЛЯ ОЦЕНКИ РАБОТЫ АЛГОРИТМА

Ш.Жўраев

Андижанский педогогический институт

С. Жўраева

Андижанский район 37- учитель школы

В практике широко используются различные способы оценки работы алгоритма. Анализируя алгоритм, можно стараться найти точное количество выполняемых им действий. Но в большинстве случаев достаточно оценить асимптотику роста времени работы алгоритма при стремлении размера входа к бесконечности (asymptotic efficiency). Если у одного алгоритма асимптотика роста меньше, чем у другого, то в большинстве случаев он будет эффективнее для всех входов, кроме совсем коротких[1,2]. Для оценки работа алгоритма, в практике широко используются асимптотические обозначения.

Одним из асимптотических обозначений является Θ-обозначение.

Например, что время T(n) работы алгоритма сортировки вставками на входах длины n есть $\Theta(n^2)$. Точный смысл этого утверждения такой: найдутся такие константы c1, c2 > 0 и такое число n0, что c1n2 ≤ T(n) ≤ $c2n^2$ при всех n ≥ n0. Вообще, если g(n) — некоторая функция, то запись f(n) = Θ(g(n)) означает, что найдутся такие c1, c2 > 0 и такое n0, что 0 ≤ c1g(n) ≤ f(n) ≤ c2g(n) для всех n ≤ n0 (рис.1(а)). (Запись f(n) = Θ(g(n)) читается так: «эф от эн есть тэта от же от эн».)

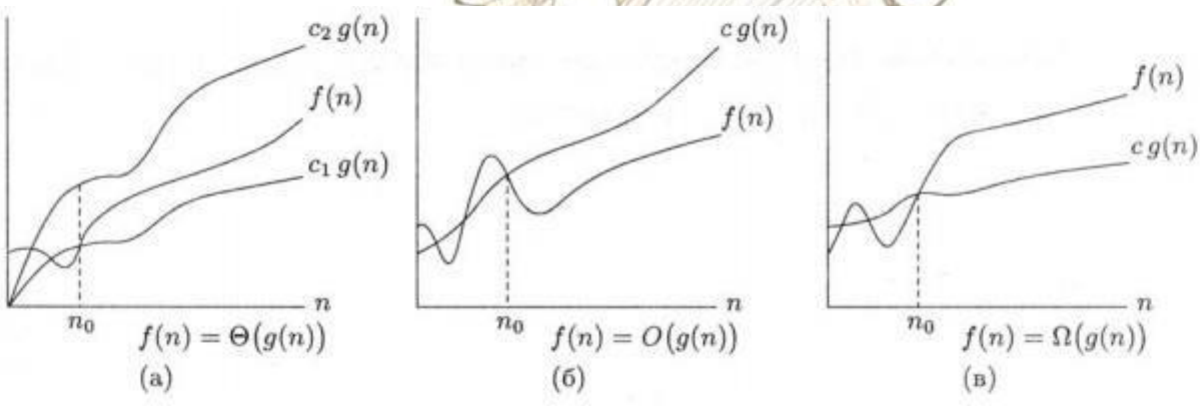

Рис. 1. Иллюстрация к определениям f(n) = Θ(g(n)), f(n) = O(g(n)) и f(n) = Ω(g(n)).

Разумеется, это обозначение следует употреблять с осторожностью: установив, что

f1(n) = Θ(g(n)) и f2(n) = Θ(g(n)), не следует заключать, что f1(n) = f2(n)!

Определение Θ(g(n)) предполагает, что функции f(n) и g(n) асимптотически неотрицательны (asymptotically nonnegative), т. е. неотрицательны для достаточно больших значений n. Заметим, что если функции f и g строго положительны, то можно исключить n0 из определения (изменив константы c1 и c2 так, чтобы для малых n неравенство также выполнялось).

Если f(n) = Θ(g(n)), то говорят, что g(n) является асимптотически точной оценкой (asymptotically tight bound) для f(n). На самом деле это отношение симметрично: если f(n) = Θ(g(n)), то g(n) = Θ(f(n)).

Допустим, что $(1/2)n^2 - 3n = \Theta(n^2)$. Согласно определению надо указать положительные константы c1, c2 и число n0 так, чтобы неравенства

$$c_1 n^2 \leq 1/2\, n^2 - 3n \leq c_2 n^2$$

выполнялись для всех $n \geq n_0$. Разделим выражение на n^2:

$$c1 \leq 1/2 - 3/n \leq c_2$$

Видно, что для выполнения второго неравенства достаточно положить c2 = 1/2. Первое будет выполнено, если (например) n0 = 7 и c1 = 1/14.

Другой пример использования формального определения: покажем, что $6n^3 \neq \Theta(n2)$. В самом деле, пусть найдутся такие c2 и n0, что $6n^3 \leq c2n^2$ для всех $n \geq n0$. Но тогда $n \leq c_2/6$ для всех $n \geq n0$ — что явно не так.

Отыскивая асимптотически точную оценку для суммы, мы можем отбрасывать члены меньшего порядка, которые при больших n становятся малыми по сравнению с основным слагаемым. Заметим также, что коэффициент при старшем члене роли не играет (он может повлиять только на выбор констант c1 и c2). Например, рассмотрим квадратичную функцию f(n) = an2 + bn + c, где a, b, c — некоторые константы и a > 0. Отбрасывая члены младших порядков и коэффициент при старшем члене, находим, что f(n) = Θ(n2). Чтобы

убедиться в этом формально, можно положить c1 = a/4, c2 = 7a/4 и n0 = 2 □ max(b / a,)

(проверьте, что требования действительно выполнены). Вообще, для любого полинома p(n) степени d с положительным старшим коэффициентом имеем p(n) = Θ(nd).

Упомянем важный частный случай использования Θ-обозначений: Θ(1) обозначает ограниченную функцию, отделённую от нуля некоторый положительной константой при достаточно больших значениях аргумента.

Кроме того, можно использовать O- и Ω-обозначения. Запись f(n) = Θ(g(n)) включает в себя две оценки: верхнюю и нижнюю. Их можно разделить. Говорят, что f(n) = O(g(n)), если найдётся такая константа c > 0 и такое число n0, что $0 \leq f(n) \leq cg(n)$ для всех $n \geq n0$ (рис.1(б)). Говорят, что f(n) = Ω(g(n)), если найдется такая константа c > 0 и такое число n0, что $0 \leq cg(n) \leq f(n)$ для всех $n \geq n0$ (рис. 1(в)). Эти записи читаются так: «эф от эн есть о большое от же от эн», «эф от эн есть омега большая от же от эн».

По-прежнему мы предполагаем, что функции f и g неотрицательны для достаточно больших значений аргумента. Легко видеть, что выполнены следующие свойства:

Теорема 1. Для любых двух функций f(n) и g(n) свойство $f(n) = \Theta(g(n))$ выполнено тогда и только тогда, когда $f(n) = O(g(n))$ и $f(n) = \Omega(g(n))$.

Для любых двух функций f(n) и g(n) свойства $f(n) = O(g(n))$ и $g(n) = \Omega(f(n))$ равносильны.

Например, $an^2+bn+c = \Theta(n^2)$ (при положительных a). Поэтому $an^2+bn+c = O(n^2)$. Другой пример: при $a > 0$ можно написать $an+b = O(n2)$ (положим $c = a+|b|$ и $n0 = 1$). Заметим, что в этом случае $an + b \neq \Omega(n^2)$ и $an + b \neq \Theta(n^2)$.

Асимптотические обозначения (Θ, O и Ω) часто употребляются внутри формул.

Например, в рекуррентном соотношении

$T(n) = 2T(n/^2) + \Theta(n)$

определено время работы сортировки слиянием. Здесь Θ(n) обозначает некоторую функцию, про которую нам важно знать лишь, что она не меньше c1n и не больше c2n для некоторых положительных c1 и c2 и для всех достаточно больших n.

Часто асимптотические обозначения употребляются не вполне формально, хотя их подразумеваемый смысл обычно ясен из контекста. Например, мы можем написать выражение

$$\sum_{i=1}^{n} O(i),$$

имея в виду сумму h(1)+h(2)+…+h(n), где h(i) — некоторая функция, для которой h(i) = O(i). Легко видеть, что сама эта сумма как функция от n есть O(n2).

Аналогичным образом вводится ω-обозначение: говорят, что f(n) есть ω(g(n)) («эф от эн есть омега малая от же от эн»), если для всякого положительного c найдется такое n0, что 0

$\leq cg(n) \leq f(n)$ при всех $n \geq n0$. Очевидно, $f(n) = \omega(g(n))$ равносильно $g(n) = o(f(n))$.

Пример: $n^2/2 = \omega(n)$, но $n^2/2 \neq \omega(n^2)$.

СПИСОК ЛИТЕРАТУРЫ:

1. Кнут Д. Э. Искусство программирования. Том 1. Основные алгоритмы = The Art of Computer Programming. Volume 1. Fundamental Algorithms / под ред. С. Г. Тригуб (гл. 1), Ю. Г. Гордиенко (гл. 2) и И. В. Красикова (разд. 2.5 и 2.6). — 3. — Москва: Вильямс, 2002. — Т. 1. — 720 с.

2. Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein. Introduction to Algorithms. Tread edition. The MIT Press Cambridge, Massachusetts London, England 2009. 1200 p.