Algorithms and Data Structures Chapter 3

Catherine Durso

cdurso@cs.du.edu

1. Algorithms and Data Structures – 1/35

Chapter 3

Chapter 3 formalizes the meaning of the notations O(g(n)) and $\Theta(g(n))$ and introduces o(g(n)), $\Omega(g(n))$ and $\omega(g(n))$. These concepts provide ways to describe how the running time of an algorithm increases with size in the limit as the size of the input increases. Considering limiting behavior allows examination of the time requirements of algorithms without implementing the algorithm and without requiring detailed knowledge of system on which the algorithm is implemented.

The chapter reviews the properties of functions frequently used as reference functions in asymptotic analysis. Some asymptotic relations of these reference functions are supplied.

Functions for Analysis

The functions discussed in the context of O(g(n)), $\Theta(g(n))$, o(g(n)), $\Omega(g(n))$ and $\omega(g(n))$ are functions whose domains are $\mathbb{N} = \{0, 1, 2, ...\}$. The definitions are most meaningful for functions that take on negative values for at most a finite set of arguments.

Used precisely, O(g(n)), $\Theta(g(n))$, o(g(n)), $\Omega(g(n))$ and $\omega(g(n))$ denote subsets the set of such functions, though in conventional use one typically writes, for example, $f = \Theta(g(n))$ rather than $f \in \Theta(g(n))$.

Motivation

equals the asymptotic growth rate of g."

"_"

"<"

"<"

"≥"

 $f(n) = O(g(n)) \approx$ "The asymptotic growth rate of f is less than or equal to the asymptotic growth rate of g." $f(n) = o(g(n)) \approx$ "The asymptotic growth rate of f is less than the asymptotic growth rate of q."

 $f(n) = \Theta(q(n)) \approx$ "The asymptotic growth rate of f

 $f(n) = \Omega(g(n)) \approx$ "The asymptotic growth rate of f is greater than or equal to the asymptotic growth rate of g."

">"

 $f\left(n\right)=\omega\left(g\left(n\right)\right)\approx\text{``The asymptotic growth rate of }f\text{ is greater than the asymptotic growth rate of }g.\text{''}$

 Θ – notation

For a given function $g\left(n\right),\ \Theta\left(g\left(n\right)\right)$ denotes the set defined by

 $\Theta(g(n)) = \{f : \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \\ \text{such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n > n_0 \} \\ \text{Intuitively, } f = \Theta(g(n)) \text{ or } f \in \Theta(g(n)) \text{ if the values} \\ \text{of } f \text{ are sandwiched between the values of two multiples} \\ \text{of } g \text{ for sufficiently large arguments.} \end{cases}$

Θ Example

If g(n) = n and f(n) = 100n + 1000, then f is in $\Theta(g(n))$. To show this, find positive constants c_1 , c_2 , and n_0 for which $0 \le c_1 n \le 100n + 1000 \le c_2 n$ for all $n > n_0$.

One possibility is $c_1 = 1$, $c_2 = 110$, and $n_0 = 100$. Then $0 \le n \le 100n + 1000 \le 100n + 10n = 110n$ for all $n > n_0$ because 10n > 10(100) = 1000 for such n.

There are many other possible triples for c_1 , c_2 , and n_0 .

Θ and limits

If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = l > 0$ then $f = \Theta(g(n))$, though the converse is not true.

The definition of the limit says that if $lim_{n\to\infty}\frac{f(n)}{g(n)}=l>0$, then for any $\varepsilon>0$, there exists N such that n > N implies $l - \varepsilon \leq \frac{f(n)}{q(n)} \leq l + \varepsilon$. Take $\varepsilon = \frac{l}{2}$ to get n > N implies $\frac{l}{2} \leq \frac{f(n)}{q(n)} \leq \frac{3l}{2}$ and $\frac{l}{2}g(n) \leq f(n) \leq \frac{3l}{2}g(n)$. Thus the triple $c_1 = \frac{l}{2}$, $c_2 = \frac{3l}{2}$, and $n_0 = N$ certifies that $f = \Theta(g(n))$. Can you think of a counterexample to the converse?

limit payoff

The limit method of verifying $f = \Theta(g(n))$ simplifies the demonstration that if f(n) is any polynomial $\sum_{i=0}^{d} a_i n^i$ with $a_d > 0$ and $g(n) = n^d$ then $f = \Theta(g(n))$:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \sum_{i=0}^{d} a_i n^{i-d} = a_d > 0$$

O- notation

The O-notation is useful in specifying the f grows no faster than g:

 $O\left(g\left(n\right)\right) =$

{f: there exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n > n_0$ }

O and limits

If
$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = l < \infty$$
 then $f = O(g(n))$.
If $\left\{\frac{f(n)}{g(n)} : n \in \mathbb{N}\right\}$ is bounded, then $f = O(g(n))$.

Applications of O

- If f is a polynomial of degree less than or equal to d then $f=O\left(n^{d}\right).$
- if $f(n) = \Theta(g(n))$ then f(n) = O(g(n)).
- *O*-bounds can often be determined from the depth of nesting in an iterative algorithm.
- O- or Θ- bounds on the worst case running time of an algorithm are O-bounds for the running time on arbitrary inputs.

Ω -notation

The notation $f(n) = \Omega(g(n))$ says that g provides an asymptotic lower bound on f.

 $\begin{aligned} \Omega\left(g\left(n\right)\right) &= \\ \left\{f: \text{there exist positive constants } c \text{ and } n_{0} \\ \text{such that } 0 &\leq cg\left(n\right) \leq f\left(n\right) \text{ for all } n > n_{0} \right\} \end{aligned}$

O, Θ , Ω **Theorem**

Theorem 1 For any two functions f(n) and g(n), $f(n) = \Theta(g(n))$ if and only if f(n) = O(g(n)) and $f(n) = \Omega(g(n))$.

o- notation

The o- notation f(n) = o(g(n)) indicates that g(n) grows properly faster, asymptotically, than f(n).

 $o\left(g\left(n\right)\right) =$

 $\{f : \text{ for any positive constant } c > 0 \text{ there exists a constant and } n_0 > 0 \text{ such that } 0 \le f(n) \le cg(n) \text{ for all } n > n_0 \}$

o- Example

f(n) = n and $g(n) = n^2$ satisfy f(n) = o(g(n)): for any c > 0, pick $n_0 > \frac{1}{c}$. $f(n) = n^2$ and $g(n) = n^2$ do not satisfy

 $f(n) = n^2$ and $g(n) = n^2$ do not satisfy f(n) = o(g(n)): if $c = \frac{1}{2}$, $0 \le f(n) \le cg(n)$ is true only for n = 0.

o and limits

 $\begin{array}{l} o\left(g\left(n\right)\right) = \\ \left\{f\left(n\right):f\left(n\right) \text{ is asymptotically nonnegative and} \\ lim_{n \to \infty} \frac{f(n)}{g(n)} = 0 \end{array} \right\} \end{array}$

ω -notation

The ω - notation $f(n) = \omega(g(n))$ indicates that f(n) grows properly faster, asymptotically, than g(n).

 $\omega\left(g\left(n\right)\right) =$

{ f : for any positive constant c > 0 there exists a constant $n_0 > 0$ such that $0 \le cg(n) \le f(n)$ for all $n > n_0$ }

ω - Example

 $f(n) = 4^n \text{ and } g(n) = 2^n \text{ satisfy } f(n) = \omega(g(n))$: for any c > 0, pick $n_0 > \log_2(c)$. Then, for $n > n_0$, $4^n = 2^n 2^n > 2^{n_0} 2^n > c 2^n$.

ω and limits

$$\begin{split} &\omega\left(g\left(n\right)\right) = \\ &\left\{f\left(n\right):f\left(n\right) \text{ is asymptotically nonnegative and} \\ &\lim_{n\to\infty}\frac{g(n)}{f(n)} = 0\right\} \end{split}$$

Properties

The comparisons Θ , O, Ω , o, and ω for functions are analogous to $=, \leq, \geq, <$, and > for real numbers. This analogy provides an easy way to remember some of the properties of Θ , O, Ω , o, and ω .

Transitivity: If f(n) and g(n) are asymptotically nonnegative, Θ , O, Ω , o, and ω are transitive, e.g. $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ implies $f(n) = \Theta(h(n))$.

How does a proof go?

Properties, cont.

Reflexivity: Θ , O, and Ω are reflexive relations in that $f(n) = \Theta(f(n))$ f(n) = O(f(n)) $f(n) = \Omega(f(n))$ Symmetry: $f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n))$

Trichotomy

Unlike =,<, and > for real numbers, there are pairs of asymptotically nonnegative functions f(n) and g(n) for which $f(n) \notin \Theta(g(n))$, $f(n) \notin O(g(n))$, and $f(n) \notin \Omega(g(n))$.

Consider
$$f(n) = \begin{cases} 1 & n \mod 2 = 0 \\ n^2 & n \mod 2 = 1 \end{cases}$$
 and $g(n) = n$.

Notation Extension

A statement like $T(n) = 2T(\frac{n}{2}) + o(n)$ is interpreted to mean that there is a function $f(n) \in o(n)$ for which $T(n) = 2T(\frac{n}{2}) + f(n)$.

Function basics

A function f(n) is monotonically increasing if $m \le n \Rightarrow f(m) \le f(n)$.

A function f(n) is monotonically decreasing if $m \le n \Rightarrow f(m) \ge f(n)$.

A function f(n) is strictly increasing if $m < n \Rightarrow f(m) < f(n)$.

A function f(n) is strictly decreasing if $m < n \Rightarrow f(m) > f(n)$.

floor and ceiling

For $x \in \mathbb{R}$, $\lfloor x \rfloor = \max \{z \in \mathbb{Z} | z \le x\}$ and $\lceil x \rceil = \min \{z \in \mathbb{Z} | z \ge x\}.$ $x - 1 \le \lfloor x \rfloor \le x \le \lceil x \rceil \le x + 1$ $n \in \mathbb{Z} \Rightarrow \lfloor n/2 \rfloor + \lceil n/2 \rceil = n$ For any positive real number x, and any positive integers a and b, $\lceil \frac{\lceil x/a \rceil}{b} \rceil = \lceil \frac{\lceil x \rceil}{ab} \rceil$, and $\lfloor \frac{\lfloor x/a \rfloor}{b} \rfloor = \lfloor \frac{\lfloor x \rfloor}{ab} \rfloor$.

floor proof

$$\begin{split} \frac{x}{a} &= \lfloor \frac{x}{a} \rfloor + y \text{ for some } 0 \leq y < 1 \\ \frac{x}{ab} &= \frac{\lfloor \frac{x}{a} \rfloor}{b} + \frac{y}{b} = \lfloor \frac{\lfloor \frac{x}{a} \rfloor}{b} \rfloor + \frac{k}{b} + \frac{y}{b} \text{ for some integer} \\ 0 \leq k \leq b-1 \\ \text{Note } \frac{k}{b} + \frac{y}{b} < 1 \text{ so } \lfloor \frac{x}{ab} \rfloor = \lfloor \lfloor \frac{\lfloor \frac{x}{a} \rfloor}{b} \rfloor + \frac{k}{b} + \frac{y}{b} \rfloor = \lfloor \frac{\lfloor \frac{x}{a} \rfloor}{b} \rfloor. \\ \text{The corresponding proof for ceiling is similar.} \\ \text{One consequence is that } \begin{bmatrix} \lfloor \frac{n}{m} \\ m \end{bmatrix} \\ \ddots & /m \end{bmatrix} \text{ is between } \lfloor n/m^k \rfloor \text{ and } \\ \lceil n/m^k \rceil \text{ for the appropriate power of } m. \end{split}$$

Modular Arithmetic

For any integer a, and any integer n > 1, $a \mod n = a - n\lfloor a/n \rfloor$. If $a \mod n = b \mod n$, write $a \equiv b \pmod{n}$.

Arithmetic operations can be done before or after evaluation mod n: $(a + b) \mod n \equiv a \mod n + b \mod n \pmod{n}$, $(\mod n)$, $(ab) \mod n \equiv a \mod n \cdot b \mod n \pmod{n}$ and $n \pmod{n}$ and $a^k \mod n \equiv (a \mod n)^k \pmod{n}$ for any integer b, and any positive integer k.

Polynomials

Given a nonnegative integer d and constants

 $a_0, a_1, \ldots a_d$ with $a_d \neq 0$, $p(n) = \sum_{i=0}^d a_i n^i$ is a polynomial of degree d.

If $a_d > 0$, p(n) is asymptotically positive, and $p(n) = \Theta(n^d)$. If $f(n) = O(n^k)$ for some constant k, then f(n) is polynomially bounded.

Exponentiation

```
For all real n, m, and a > 0,

a^0 = 1

a^1 = a

a^{-1} = 1/a

(a^m)^n = a^{mn}

a^m a^n = a^{m+n}
```

Exponentiation Limits

For all real constants a > 1, and b, $lim_{n\to\infty}\frac{n^b}{a^n} = 0$ (by L'Hôpital's Rule), so $n^b = o(a^n)$.

 $e^{x} = \sum_{i=0}^{\infty} \frac{x^{i}}{i!}$ $e^{x} = \lim_{n \to \infty} \left(1 + \frac{x}{n}\right)^{n}$ $e \approx 2.71828$

Logarithms

Recall that for a and b positive, $\log_b a$ is defined to satisfy $b^{\log_b a} = a$.

Logarithms base 2 and base e are most commonly used in algorithm analysis, so the following notations are convenient.

 $lg n = log_2 n$ $ln n = log_e n$ $lg^k n = (lg n)^k$ lg lg n = lg (lg n)

Log Identities

For all real a > 0, b > 0, c > 0, and n, with logarithm bases not equal to 1, $\log_c (ab) = \log_c a + \log_c b$ $\log_b a^n = n \log_b a$ $\log_b \frac{1}{a} = -\log_b a$ $\log_b a = \frac{\log_c a}{\log_c b}$ $\log_b a = \frac{1}{\log_a b}$ $a^{\log_b c} = c^{\log_b a}$

Verifications

For $\log_b a = \frac{\log_c a}{\log_c b}$, verify that $\log_c b \log_b a = \log_c a$ by showing $c^{\log_c b \log_b a} = c^{\log_c a}$: $c^{\log_c b \log_b a} = b^{\log_b a} = a = c^{\log_c a}$.

For
$$a^{\log_b c} = c^{\log_b a}$$
, note $a^{\log_b c} = (b^{\log_b a})^{\log_b c} = b^{\log_b a \log_b c} = (b^{\log_b c})^{\log_b a} = c^{\log_b a}$.

lg Asymptotics

For any constant a > 0, $\lg^b n = o(n^a)$, i.e., all powers of $\lg n$ grow more slowly that all positive powers of n.

This can be derived from $\lim_{n\to\infty}\frac{n^b}{a^n}=0$ by replacing n by $\lg n$ and a by 2^a :

$$0 = \lim_{n \to \infty} \frac{(\lg n)}{(2^a)^{\lg n}} = \lim_{n \to \infty} \frac{\lg n}{n^a}.$$

Factorials

Define
$$n! = \begin{cases} 1 & n = 0 \\ n(n-1)! & n > 0 \end{cases}$$

One can show
 $n! = o(n^n)$
 $n! = \omega(2^n)$
 $\lg n! = \Theta(n \lg n)$

(You may use these results, unless specifically asked to prove them.)