EE360C: Algorithms

The Basics

Summer 2019

Department of Electrical and Computer Engineering
University of Texas at Austin

o-Notation

- Consider $2n^2 = O(n^2)$ and $2n^2 = O(n^3)$.
- Both are asymptotic upper bounds for $f(n) = 2n^2$.
- For $2n^2 = O(n^2)$, we can say $f(n) = 2n^2$ does not grow faster than $g(n) = n^2$. While, for $2n^2 = O(n^3)$, we can say $f(n) = 2n^2$ grows much slower than $t(n) = n^3$.
- We use o-notation to refer to upper bounds that are loose.

Definition 5

Given $g(n)$, we denote by $o(g(n))$ the set of functions:

$$o(g(n)) = \{f(n) : \text{for any positive constant } c > 0 \text{, there exists a constant } n_0 > 0 \text{ such that } 0 \leq f(n) < cg(n) \text{ for all } n \geq n_0\}$$

ω-Notation

- In contrast, we define ω-notation similarly to refer to lower bounds that are loose.

Definition 6

Given $g(n)$, we denote by $\omega(g(n))$ the set of functions:

$$\omega(g(n)) = \{f(n) : \text{for any positive constant } c > 0 \text{, there exists a constant } n_0 > 0 \text{ such that } 0 \leq cg(n) < f(n) \text{ for all } n \geq n_0\}$$

Summary of Asymptotic Bounds

Definition	$\boxed{?}\ c \geq 0$	$\boxed{?}\ n_0 \geq 0$	$f(n) \boxed{?} c \cdot g(n)$
$O(\cdot)$	$\exists$	$\exists$	$\leq$
$o(\cdot)$	$\forall$	$\exists$	$<$
$\Omega(\cdot)$	$\exists$	$\exists$	$\geq$
$\omega(\cdot)$	$\forall$	$\exists$	$>$

The Limit Theorems

Theorem

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

$f(n)$ grows slower than $g(n)$.

Theorem

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

$f(n)$ grows faster than $g(n)$

The Limit Theorems (contd.)

Theorem

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \text{ (where } 0 \leq c < \infty) \implies f(n) = O(g(n))$$

Theorem

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \text{ (where } 0 < c \leq \infty) \implies f(n) = \Omega(g(n))$$

Theorem

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \text{ (where } 0 < c < \infty) \implies f(n) = \Theta(g(n))$$

$f(n)$ grows at the same rate as $g(n)$

The Limit Theorems (other direction)

Limit theorems also go in the other direction if the limit exists. E.g.:

Theorem

$$f(n) = O(g(n)) \implies \lim_{n\to\infty} \frac{f(n)}{g(n)} = c,$$

(where $0 < c < \infty$), if the limit exists.

Properties of Asymptotic Growth Rates

Transitivity

$$
\begin{aligned}
f(n) = O(g(n)) \land g(n) = O(h(n)) &\implies f(n) = O(h(n)) \\
f(n) = \Omega(g(n)) \land g(n) = \Omega(h(n)) &\implies f(n) = \Omega(h(n)) \\
f(n) = \Theta(g(n)) \land g(n) = \Theta(h(n)) &\implies f(n) = \Theta(h(n))
\end{aligned}
$$

Proof

Given, for some constant $c > 0$ and $n_0 > 0$, we have
$f(n) \leq cg(n),\ \forall n \geq n_0$.
Also, $c' > 0$ and $n_0' > 0$, we have $g(n) \leq c'h(n),\ \forall n \geq n_0'$.
So, $\forall n \geq \max(n_0, n_0')$, we have $f(n) \leq cg(n) \leq cc'h(n)$
$\implies f(n) \leq \tilde{c}h(n)$, where $\tilde{c} = cc' > 0$.
$\implies f(n) = O(h(n))$.

Properties of Asymptotic Growth Rates (contd.)

Reflexivity

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

Symmetry

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

Transpose Symmetry

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

Properties of Asymptotic Growth Rates (contd.)

Sum of Functions

Suppose that $f(n)$ and $g(n)$ are such that $f(n) = O(h(n))$ and $g(n) = O(h(n))$, then $f(n) + g(n) = O(h(n))$

In general

Let k be a fixed constant, and let $f_1, f_2, \ldots, f_k$ and h be functions such that $f_i = O(h)\ \forall i$ Then $f_1 + f_2 + \cdots + f_k = O(h)$.

Properties of Asymptotic Growth Rates (contd.)

Suppose that $f(n)$ and $g(n)$ (taking non-negative values) are such that $g(n) = O(f(n))$, then $f(n) + g(n) = \Theta(f(n))$

Proof

$\forall n \geq 0$, we have $f(n) + g(n) \geq f(n)$.

Therefore,

$$f(n) + g(n) = \Omega(f(n)) \tag{1}$$

Also, $g(n) = O(f(n))$, and $f(n) = O(f(n))$ *(given, and reflexivity)*

Therefore, by sum of functions property,

$$f(n) + g(n) = O(f(n)) \tag{2}$$

From 1 and 2, we can say that $f(n) + g(n) = \Theta(f(n))$

Analogies to Traditional Comparisons

Analogies between the asymptotic comparison of two functions f and g and the comparison of two real numbers a and b.

$$\begin{aligned} f(n) = O(g(n)) &\approx a \leq b \\ f(n) = \Omega(g(n)) &\approx a \geq b \\ f(n) = \theta(g(n)) &\approx a = b \\ f(n) = o(g(n)) &\approx a < b \\ f(n) = \omega(g(n)) &\approx a > b \end{aligned}$$

The analogy does break down in some cases. For two functions $f(n)$ and $g(n)$, it may be the case that neither $f(n) = O(g(n))$ nor $f(n) = \Omega(g(n))$ holds.

Standard Functions

Standard Notations

Monotonicity

- $f(n)$ is **monotonically increasing** if $m \leq n$ implies $f(m) \leq f(n)$
- $f(n)$ is **monotonically decreasing** if $m \leq n$ implies$f(m) \geq f(n)$
- $f(n)$ is **strictly increasing** if $m < n$ implies $f(m) < f(n)$
- $f(n)$ is **strictly decreasing** if $m < n$ implies $f(m) > f(n)$

Floors and Ceilings

- For any real number x, we denote the greatest integer less than or equal to x by $\lfloor x \rfloor$.
- For any real number x, we denote the least integer greater than or equal to x by $\lceil x \rceil$.

Standard Notations (cont.)

Modular Arithmetic

For any integer a and positive integer n, the value $a \mod n$ is the **remainder** of the quotient a/n.

$$a \mod n = a - \lfloor a/n \rfloor n$$

Polynomials

Given a nonnegative integer d, a **polynomial in n of degree d** is a function $p(n)$ of the form:

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

where the constants $a_0, a_1, \ldots a_d$ are the **coefficients** of the polynomial and $a_d \neq 0$

- For an asymptotically positive polynomial $p(n)$ of degree d, we have $p(n) = \Theta(n^d)$.

Exponentials

For all real $a > 0$, m, and n, these identities hold:

- $a^0 = 1$
- $a^1 = a$
- $a^{-1} = 1/a$
- $(a^m)^n = a^{mn}$
- $(a^m)^n = (a^n)^m$
- $a^m a^n = a^{m+n}$

Any exponential function with base strictly greater than 1 grows faster than any polynomial function:

$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0$$

Logarithm Notations

$$\begin{aligned}
\lg n &= \log_2 n \\
\ln n &= \log_e n \\
\lg^k n &= (\lg n)^k \\
\lg \lg n &= \lg(\lg n)
\end{aligned}$$

For all real $a > 0$, $b > 0$, $c > 0$, and n:

$$\begin{aligned}
a &= b^{\log_b a} \\
\log_c(ab) &= \log_c a + \log_c b \\
\log_b a^n &= n \log_b a \\
\log_b a &= \frac{\log_c a}{\log_c b} \\
\log_b(1/a) &= -\log_b a \\
\log_b a &= \frac{1}{\log_a b} \\
a^{\log_b c} &= c^{\log_b a}
\end{aligned}$$

Bounds for Common Functions

Polynomials

If $f = a_0 + a_1 n + \ldots + a_d n^d$ is a polynomial of degree d, in which $a_d > 0$, then $f = O(n^d)$.

Proof

Note, a_j for $j < d$ may be negative.
But, $a_j n^j \leq |a_j| n^d \ \forall n \geq 1$
Thus, each of the term of f is $O(n^d)$
Since, f is a constant sum of functions, each of which is $O(n^d)$, it follows f is $O(n^d)$

Similarly, $f = \Omega(n^d)$ and $f = \Theta(n^d)$

Polynomials (contd.)

Polynomial Time Algorithm

An algorithm whose running time $T(n)$ is $O(n^d)$ for some constant d (where d is independent of the input size).

Note:

An algorithm can be polynomial time even if the running time is of the form $O(n^x)$, where x is not an integer.

Logarithms

If $x = \log_b n$, then $b^x = n$.

Logarithms grow slower than polynomials. i.e.

$$\lim_{n \to \infty} \frac{\log n}{n^d} = 0$$

For every $b > 1$ and every $x > 0$, we have $\log_b n = O(n^x)$

$O(\log_a n) = O(\log_b n)$ for any constants $a, b > 0$. (You can ignore the base in logarithms.)

Exponentials

Exponentials

For every $r > 1$ and every $d > 0$, $n^d = O(r^n)$. (Every exponential grows faster than every polynomial.)

Questions?