Australian National University

COMP3600/6466 Algorithms

Lecture 3

S2 2016

Dr. Hassan Hijazi

Prof. Weifa Liang

Asymptotic notations

$O(g(n))= \{f(n) : \text{there } \textbf{exist} \text{ positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le c \cdot g(n) \text{ for } \textbf{all } n \ge n_0\}$, e.g. $8 \lg n = O(n)$.

$\Omega(g(n))= \{f(n) : \text{there } \textbf{exist} \text{ positive constants } c \text{ and } n_0 \text{ such that } 0 \le c \cdot g(n) \le f(n) \text{ for } \textbf{all } n \ge n_0\}$, e.g. $26n^7 + 2013 = \Omega(n^5)$.

$\Theta(g(n))= \{f(n) : \text{there } \textbf{exist} \text{ positive constants } c_1, c_2 \text{ and } n_0 \text{ such that } 0 \le c_1 \cdot g(n) \le f(n) \le c_2 \cdot g(n) \text{ for } \textbf{all } n \ge n_0\}$, e.g. $5 \cdot 2^n + n^6 - 10^{10} = \Theta(2^n)$.

$o(g(n))= \{f(n) : \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \ (f(n) \text{ and } g(n) \text{ positive})\}$, e.g. $10^5 \cdot 2^n = o(3^n)$.

$\omega(g(n))= \{f(n) : \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty \ (f(n) \text{ and } g(n) \text{ positive})\}$, e.g. $2^n = \omega(n^3)$.

Asymptotic notations

$$f(n) = O(g(n))$$

$$\Updownarrow$$

There **exist positive** c and n_0 such that $0 \leq f(n) \leq c \cdot g(n)$ for **all** $n \geq n_0$.

$$\Downarrow$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} \leq c$$

If the limit exists

$$\lim_{n\to\infty} \left(\frac{1}{n} + 1\right) \leq 1 \not\Rightarrow \left(\frac{1}{n} + 1\right) \leq 1$$

$$\lim_{n\to\infty} \left(\frac{f(n)}{g(n)}\right) = c \implies f(n) = \Theta(g(n)) \left(c_1 = \frac{c}{2} \text{ and } c_2 = \frac{3}{2}c\right)$$

Order of growth hierarchy

• Constant	c
• Logarithmic	$\lg n$
• Fractional Power	n^{ϵ}
• Linear	n
• Quasilinear	$n \lg^k n$
• Polynomial	n^k
• Quasi-polynomial	$2^{\lg^k n}$
• Exponential	2^{n^k}
	$(0 < \epsilon < 1)$
	$(k \geq 1)$

Increasing order

$$g_1(n)$$
$$g_2(n)$$
$$\dots$$
$$g_k(n)$$

$$g_i(n) = O(g_{i+1}(n)),\ \forall i \in \{1, \dots, k-1\}$$

OR

$$g_i(n) = \Omega(g_{i-1}(n)),\ \forall i \in \{2, \dots, k\}$$

Asymptotic notation in equations

Consider the statement: $af(n) = O(f(n)), a > 0$

$O(f(n))$ is a set

$O(f(n))$ is a set of infinite size

Asymptotic notation in equations

Reminder: $f(n) = O(g(n))$ stands for $f(n) \in O(g(n))$

$$2n^3 + 3n^2 + 5 = 2n^3 + \Theta(n^2)$$

Interpretation: There is some function $f(n)$ in $\Theta(n^2)$ such that $2n^3 + 3n^2 + 5 = 2n^3 + f(n)$.

$$2n^3 + \Theta(n^2) = \Theta(n^3)$$

Interpretation: For any choice $f(n) \in \Theta(n^2)$, there is a function $g(n) \in \Theta(n^3)$ such that $2n^3 + f(n) = g(n)$.

Asymptotic notation in equations

Transitivity holds for O-notation:
$f(n) = O(g(n))$ and $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$.

It also holds for Ω, Θ, o, and ω.

Examples:

1. $\frac{1}{n} = O(\log n)$ and $\log n = O(n^3) \Rightarrow \frac{1}{n} = O(n^3)$
2. $3n^3 + 2n^2 + n = \Omega(3n^3)$ and $3n^3 = \Omega(n^3) \Rightarrow 3n^3 + 2n^2 + n = \Omega(n^3)$
3. $2n^2 = \Theta(n^2)$ and $n^2 = \Theta(3n^2) \Rightarrow 2n^2 = \Theta(3n^2)$

where the symbol $\Rightarrow$ means "implies".

Asymptotic notation in equations

Sample proof for transitivity

Claim: $f(n) = O(g(n))$ and $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$.

Proof:

- $f(n) = O(g(n))$ means:
  There are constants $n_1 > 0$ and $c_1 > 0$ such that $0 \leq f(n) \leq c_1 g(n)$ for $n \geq n_1$.

- $g(n) = O(h(n))$ means:
  There are constants $n_2 > 0$ and $c_2 > 0$ such that $0 \leq g(n) \leq c_2 h(n)$ for $n \geq n_2$.

- We obtain $0 \leq f(n) \leq c_1 g(n) \leq c_1 c_2 h(n)$ for $n \geq \max(n_1, n_2)$.

- Since $c = c_1 c_2$ and $n_0 = \max(n_1, n_2)$ are positive constants, $f(n) = O(h(n))$ follows by the-O definition.

Asymptotic notation in equations

Comparison of functions

Let $f(n)$ and $g(n)$ be asymptotically positive.

- Reflexivity:
  1. Holds for O, Ω, and Θ, e.g. $f(n) = O(f(n))$, $4n^2 = \Omega(4n^2)$, $n \log n = \Theta(n \log n)$.
  2. Not true for $o(\,)$ and $\omega(\,)$, e.g. $\log n \neq o(\log n)$, $n \neq \omega(n)$.
- Symmetry:
  1. $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$.
  2. But this is not true for O and Ω, e.g. $\log \log n = O(\log n)$ but $\log n \neq O(\log \log n)$.

Asymptotic notation in equations

Sample proof for reflexivity

Proof:

- We want to show $f(n) = \Omega(f(n))$, which means:
  There are constants $n_0 > 0, c > 0$ such that $0 \leq cf(n) \leq f(n)$ for $n \geq n_0$.

- This is true, because $1 \cdot f(n) \leq f(n)$ for all natural numbers, and $f(n)$ is assumed to be asymptotically nonnegative.

- More precisely, $c = 1$ and some large enough n_0 satisfy the definition, so $f(n) = \Omega(f(n))$ follows.

Asymptotic notation in equations

Further properties

- $O(g_1(n) + g_2(n)) = O(\max\{g_1(n), g_2(n)\})$, e.g. $O(n^5 + \log n) = O(n^5)$.
- $O(g_1(n)) + O(g_2(n)) = O(\max\{g_1(n), g_2(n)\})$, e.g. $O(n^3) + O(n^2) = O(n^3)$.
- $O(g_1(n)) \cdot O(g_2(n)) = O(g_1(n) \cdot g_2(n))$, e.g. $O\left(\frac{1}{n}\right) \cdot O(\log n) = O\left(\frac{\log n}{n}\right)$.
- However, $\dfrac{O(n^2)}{O(n)}$ is meaningless.
- $\dfrac{\Omega(g_1(n))}{O(g_2(n))} = \Omega\left(\dfrac{g_1(n)}{g_2(n)}\right)$, e.g. $\dfrac{1}{O(n)} = \dfrac{\Omega(1)}{O(n)} = \Omega\left(\dfrac{1}{n}\right)$.
- $\dfrac{O(g_1(n))}{\Omega(g_2(n))} = O\left(\dfrac{g_1(n)}{g_2(n)}\right)$, e.g. $\dfrac{O(n^2)}{\Omega(n)} = O(n)$.

Notations and Common Functions

- Monotonicity: A function $f(n)$ is monotonically increasing if $m < n$ implies $f(m) \leq f(n)$ and it is strictly increasing if $m < n$ implies $f(m) < f(n)$. Similarly, $f(n)$ is monotonically decreasing if $m < n$ implies $f(m) \geq f(n)$ and strictly decreasing if $m < n$ implies $f(m) > f(n)$.

- Floors and ceilings: For any real x, the floor of x, denoted by $\lfloor x \rfloor$, is the greatest integer less than or equal to x. The ceiling of x, denoted by $\lceil x \rceil$, is the least integer greater than or equal to x. We have $x - 1 < \lfloor x \rfloor \leq x \leq \lceil x \rceil < x + 1$.

- Polynomials: Given a nonnegative integer d, a polynomial in n is a function of the form $p(n) = \sum_{i=0}^{d} c_i n^i$, where c_i is a constant for each i, called a coefficient of the polynomial. The degree of a polynomial is the highest power that occurs.
  E.g. $p(n) = 3n^7 - 4n^5 + 1.1n^2 - 100$ is a polynomial of degree 7.

Notations and Common Functions

Iterated logarithm

$\lg^{(0)} n = n$, $\lg^{(1)} n = \lg n$, $\lg^{(2)} n = \lg\lg n$, $\lg^{(3)} n = \lg\lg\lg n, \ldots$
In general: $\lg^{(k+1)} n = \lg\lg^{(k)} n$.
Iterated logarithm function: $\lg^* n = \min\{i \geq 0 \mid \lg^{(i)} n \leq 1\}$.

Exercise: Determine the value of $\lg^* 2^{16}$.

Solution: $\lg^* 2^{16}$ is the smallest integer i such that $\lg^{(i)} 2^{16} \leq 1$.
We have

$$
\begin{aligned}
\lg^{(0)} 2^{16} &= 2^{16} && >1\\
\lg^{(1)} 2^{16} &= \lg 2^{16} = 16 && >1\\
\lg^{(2)} 2^{16} &= \lg\lg^{(1)} 2^{16} = \lg 16 = \lg 2^4 = 4 && >1\\
\lg^{(3)} 2^{16} &= \lg\lg^{(2)} 2^{16} = \lg 4 = \lg 2^2 = 2 && >1\\
\lg^{(4)} 2^{16} &= \lg\lg^{(3)} 2^{16} = \lg 2 = 1 && \leq 1.
\end{aligned}
$$

It follows that $\lg^* 2^{16} = 4$.

Notations and Common Functions

- Exponentials: $(a^m)^n = a^{m \cdot n}$, $a^m a^n = a^{m+n}$, $\lim_{n\to\infty}(1+\frac{x}{n})^n = e^x$, $e^{\ln x} = \ln(e^x) = x$.

- Logarithms: $\lg n = \log_2 n$, $\ln n = log_e n$, $\lg^k n = (\lg n)^k$, $\lg\lg n = \lg(\lg n)$.

- Factorials: Recall the definition of n factorial: $n! = 1 \cdot 2 \cdot 3 \cdots (n-1) \cdot n$.
  We have $n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$. (Stirling's formula)

Exercise 2.6

For each statement, select between

always true, never true, or sometimes true:

1. $f(n) = O(f(n)^2)$

2. $f(n) = \Omega(g(n))$ and $f(n) = o(g(n))$

Exercise 3.1

Let $f(n) = \sum_{i=1}^{n} i$ and $g(n) = n^2$. Prove that $f(n) = \Theta(g(n))$

Proof: First we rewrite $f(n)$ in the form $f(n) = \frac{n(n+1)}{2} = \frac{n^2+n}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$.
We have to determine constants $c_1, c_2 > 0$ such that

$$0 \leq c_1 \cdot n^2 \leq \frac{1}{2}n^2 + \frac{1}{2}n \leq c_2 \cdot n^2,$$

or, equivalently,

$$0 \leq c_1 \leq \frac{1}{2} + \frac{1}{2n} \leq c_2,$$

when n is sufficiently large.
Since $\frac{1}{2n}$ can be made arbitrarily small by choosing large enough values of n, $c_1 = \frac{1}{4}$ and $c_2 = \frac{3}{4}$ satisfy the above inequalities for all sufficiently large n.
Thus,

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

Exercise 3.2

$$O(n^2) \neq O(n)$$

Proof: If $O(n^2) = O(n)$ was true, then, for any choice $f(n)$ from the set $O(n^2)$, there would be a function $g(n)$ in $O(n)$ such that $f(n) = g(n)$. In other words, $O(n^2)$ would be a subset of $O(n)$. (Here, it is precise to use "subset", because on both sides of the equation, we have well-defined sets.)

However, $f(n) = n^2$ is a member of $O(n^2)$ but it is not a member of $O(n)$. Thus, $O(n^2)$ is not a subset of $O(n)$, that is, $O(n^2) \neq O(n)$. (Note that there are many many other functions in $O(n^2)$ that are not in $O(n)$.)

Exercise 3.3

$$n + n^2 O(\ln n) = O(n^2 \ln n)$$

Proof: We have to prove that, for any choice $f(n)$ from the set $O(\ln n)$, there is a function $g(n)$ in $O(n^2 \ln n)$ such that $n + n^2 f(n) = g(n)$, that is, $n + n^2 f(n)$ is an element of $O(n^2 \ln n)$.

Let $f(n)$ be an arbitrary element of $O(\ln n)$. This means that, there is some $c > 0$ constant such that $0 \leq f(n) \leq c \cdot \ln n$ for all sufficiently large values of n.

We obtain

$$0 \leq n^2 f(n) \leq c \cdot n^2 \ln n,$$

$$(0 \leq\ ) \ n \leq n + n^2 f(n) \leq n + c \cdot n^2 \ln n \leq n^2 \ln n + c \cdot n^2 \ln n \leq (c+1) \cdot n^2 \ln n$$

for large n. By the definition of O, this means that $n + n^2 f(n) = O(n^2 \ln n)$, as required.

Exercise 3.4

Let $p(n) = \sum_{i=0}^{d} a_i n^i$ where $a^d > 0$. $p(n)$ is a degree-d polynomial in n. Given a constant k, prove the following properties:

1. If $k \geq d$, then $p(n) = O(n^k)$.
2. If $k \leq d$, then $p(n) = \Omega(n^k)$.
3. If $k = d$, then $p(n) = \Theta(n^k)$.
4. If $k > d$, then $p(n) = o(n^k)$.
5. If $k < d$, then $p(n) = \omega(n^k)$.

Exercise 3.5

Explain why the following statement is meaningless:

"The running time of algorithm A is at least $O(n^2)$"

Exercise 3.6

Show that $2^{n+1} = O(2^n)$, but $2^{2n} \neq O(2^n)$

Exercise 3.7

Show that $\lceil \lg n \rceil!$ is not polynomially bounded, but $\lceil \lg \lg n \rceil!$ is.