Practice with Big-Oh notation

William Hendrix

Lecture 5

Today

- Review
  - Formal definitions
  - Properties

- Logarithms review

Formal definitions

$f(n) = O(g(n))$ if and only if there exist positive constants c and n_0 such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

$f(n) = \Omega(g(n))$ if and only if there exist positive constants c and n_0 such that $f(n) \geq cg(n)$ for all $n \geq n_0$.

$f(n) = \Theta(g(n))$ if and only if there exist positive constants c_1, c_2, and n_0 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.

- *Analogy:* O, Ω, and Θ act like $\leq$, $\geq$, and $=$
- Most algorithms we discuss will belong to a few classes:
  $O(1) \ll O(\lg n) \ll O(n) \ll O(n \lg n) \ll O(n^2) \ll O(n^3) \ll O(2^n) \ll O(n!)$

Properties of Big-Oh notation

- Transitivity
  $f(n) = O(g(n))$ and $g(n) = O(h(n)) \rightarrow f(n) = O(h(n))$
  $f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n)) \rightarrow f(n) = \Omega(h(n))$
  $f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \rightarrow f(n) = \Theta(h(n))$
- Equivalence rules
  $f(n) = O(g(n)) \Leftrightarrow g(n) = \Omega(f(n))$
  $f(n) = O(g(n))$ and $f(n) = \Omega(g(n)) \Leftrightarrow f(n) = \Theta(f(n))$
- Reflexivity and symmetry
  $f(n) = O(f(n))$, $f(n) = \Omega(f(n))$, and $f(n) = \Theta(f(n))$
  $f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$
- All three ignore constant coefficients
  $\forall x > 0, xf(n) = O(f(n))$, $xf(n) = \Omega(f(n))$, and
  $xf(n) = \Theta(f(n))$
- Only the largest term matters
  $f(n) = O(g(n)) \rightarrow O(f(n) + g(n)) = O(g(n))$
  $f(n) = O(g(n)) \rightarrow \Omega(f(n) + g(n)) = \Omega(g(n))$
  $f(n) = O(g(n)) \rightarrow \Theta(f(n) + g(n)) = \Theta(g(n))$

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

1. $n \lg n \neq \Omega(n^2)$

2. $\sum_{i=1}^{n} i = O(n^2)$

3. For any $x > 0$, if $f(n) = \Theta(g(n))$, then $xf(n) = \Theta(g(n))$

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

1. *Proof.* We show that for any positive constants c and n_0, there is some $n \geq n_0$ such that $n \lg n > cn$. (Apply De Morgan's Laws to the formal definition *carefully*.)
Let c and n_0 be positive constants. Consider $n = \max\{2^c, n_0\} + 1$. Note that $n \geq n_0$. Since $n \geq 2^c + 1$, $n \lg n \geq n \lg(2^c + 1)$. Since $\lg n$ is always increasing, $n \lg(2^c + 1) > n \lg(2^c) = nc$. Thus, there is some $n \geq n_0$ such that $n \lg n > cn$. □

2. *Proof.* Consider the sum $\sum_{i=1}^{n} i$, which equals $1 + 2 + \ldots + n$. Note that every term in this sum is at most n. So, $\sum_{i=1}^{n} i \leq n+n+\ldots+n = n(n) = n^2$. Since $\sum_{i=1}^{n} i \leq n^2$, there exist positive constants $c = n_0 = 1$ such that $\sum_{i=1}^{n} i \leq cn^2$ for all $n \geq n_0$, so $\sum_{i=1}^{n} i = O(n^2)$. □

Big-Oh exercises

- Use the *formal definitions* of Big-Oh, Big-Omega, and Big-Theta to prove the following:

3. *Proof.* Since $f(n) = \Theta(g(n))$, there exist positive constants c and n_0 such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$, for all $n \geq n_0$. Since $x > 0$, we may multiply all sides of this inequality by x, yielding $xc_1 g(n) \leq xf(n) \leq xc_2 g(n)$. Thus, there exist constants $c_3 = xc_1$, $c_4 = xc_2$, and n_0 such that $c_3 g(n) \leq xf(n) \leq c_4 g(n)$ for all $n \geq n_0$, so $xf(n) = \Theta(g(n))$. □

One more property

- Envelopment
  - Addition

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

  - Multiplication

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

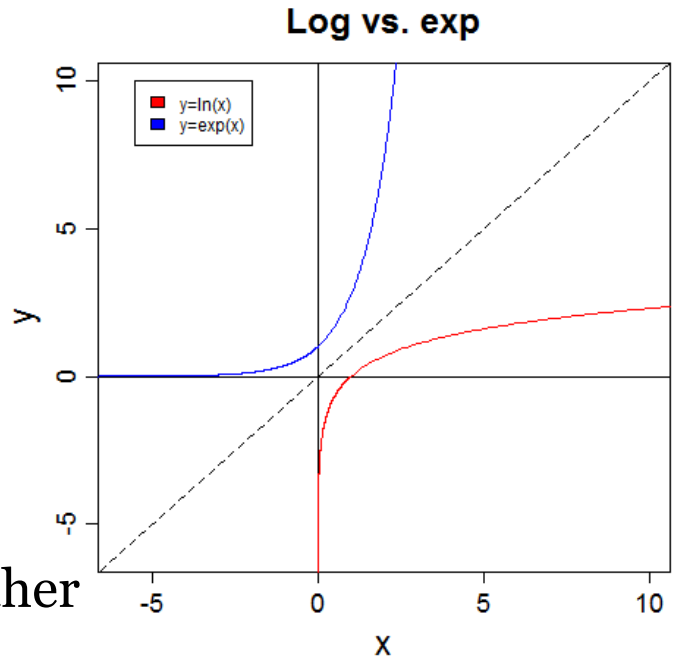
- Informally, you can move the entire expression inside Big-Oh

Revenge of the logarithms

- **Logarithm:** inverse exponential function

$$y = \ln x \Leftrightarrow x = e^y$$

- Natural log (ln): inverse of e^x
- Logarithms of other base: $\log_b(x)$
  - $\log_2(x)$ is very common in algorithms
- Computing logs of other bases
  - $\log_b(x) = \frac{\ln x}{\ln b}$
  - All logs are *scalar multiples* of one another



- **Log properties**

	Because
Base 2 → $\lg(ab) = \lg(a) + \lg(b)$	$2^A 2^B = 2^{A+B}$
$\lg(a^b) = b\lg(a)$	$(2^A)^b = 2^{Ab}$
$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\lg n)$	$\int_1^n \frac{1}{x} dx = \ln n$
$\lg(n!) = \Theta(n \lg n)$	Properties 1 and 3 above

Coming up

- More Big-Oh practice

- **Homework 3** is due tonight
- **Homework 4** is due Thursday
- **Homework 5** is *delayed*

- **Recommended readings:** Chapter 2
- **Practice problems:** attempt 1-2 problems from "Interview Problems" (p. 63)