CSC110 Lecture 21: Asymptotic Notation for Function Growth

David Liu, Department of Computer Science

*Navigation tip for web slides: press **?** to see keyboard navigation controls.*

Announcements and today’s plan

Test 2 done!

Announcements

- Assignment 3 has been posted—**due tomorrow!**
  - Check out the A3 FAQ (+ corrections)
  - Additional TA office hours
  - Review advice on academic integrity
- Next week is **reading week**!
  - No lecture, tutorial, or office hours

Story so far: evaluating programs

What makes a “good” program?

1. Correctness
2. Design and code style

3. Efficiency, or how long a program takes to run

But what does it mean to say that one program is “more efficient” than another?

Today you'll learn to...

1. Define and explain the differences between Big-O, Omega, and Theta asymptotic bounds.
2. Prove statements involving asymptotic notation.
3. Compare different elementary functions using asymptotic notation.

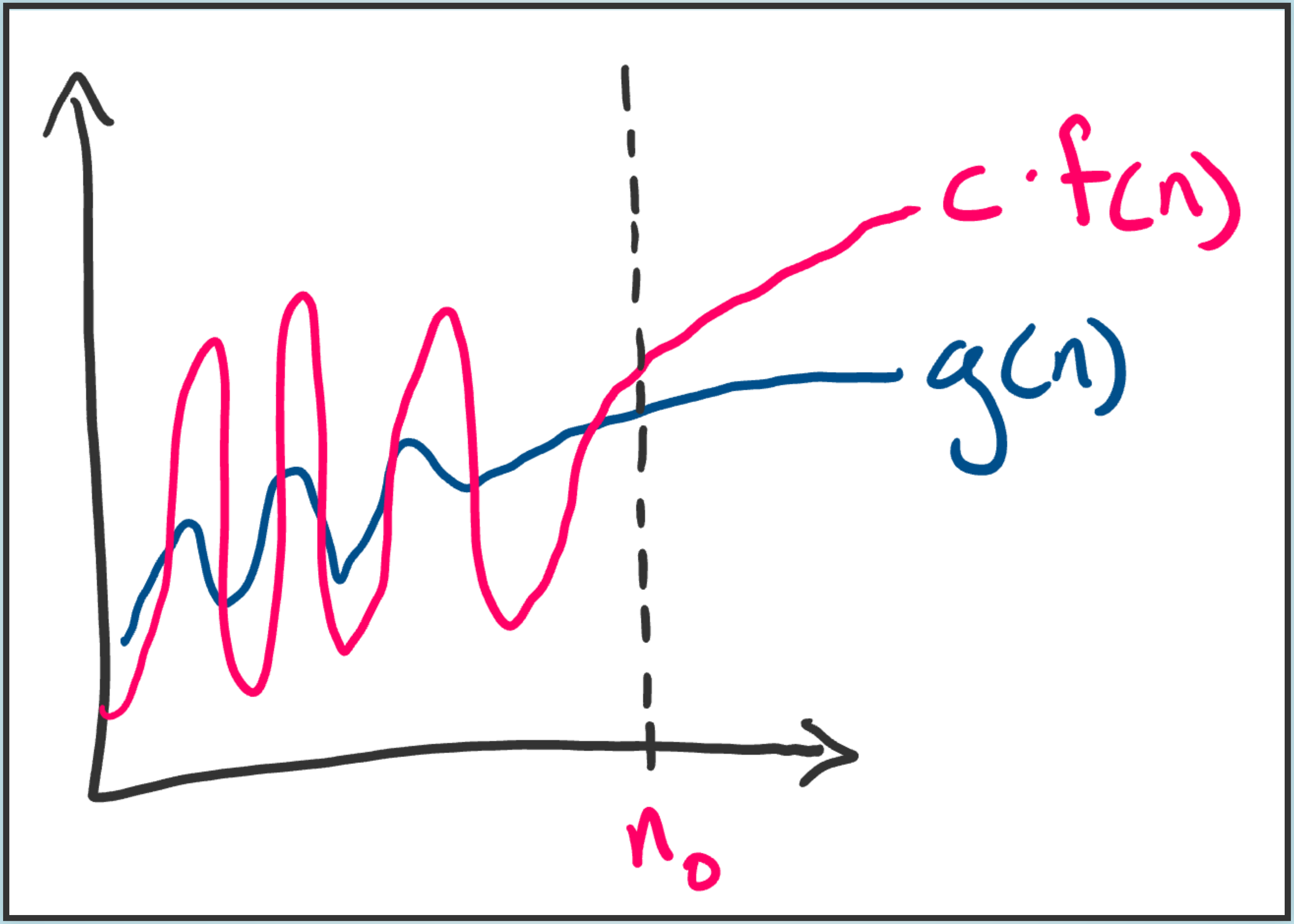
Big-O Notation

Definition of Big-O

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. We say g **is Big-O of** f and write $g \in \mathcal{O}(f)$ when:

$$\exists c, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \leq c \cdot f(n)$$

Equivalently, "g is eventually dominated up to a constant factor by f"


c · f(n)
g(n)
n0

An example

Prove that for all $a, b \in \mathbb{R}^+$, $a + bn \in \mathcal{O}(n^2)$.

(Example: $1 + 10^{10^{10}} n \in \mathcal{O}(n^2)$)

Translation:

$$\forall a, b \in \mathbb{R}^+,\ \exists c, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow a + bn \leq cn^2$$

Proof (header)

Let $a, b \in \mathbb{R}^+$.

Let $c = \ldots$ and let $n_0 = \ldots$

Let $n \in \mathbb{N}$ and assume $n \geq n_0$.

We'll prove that $a + bn \leq cn^2$.

Rough work: $a + bn \leq cn^2$

Key idea: split up into two simpler inequalities,

$$\begin{aligned} a &\leq c_1 n^2 \\ bn &\leq c_2 n^2 \end{aligned}$$

(Adding these two inequalities yields $a + bn \leq (c_1 + c_2)n^2$.)

Approach 1: Focus on "c"

$$
\begin{aligned}
a &\le c_1 n^2 \\
bn &\le c_2 n^2
\end{aligned}
$$

Pick c_1 and c_2 to satisfy inequalities.

Assuming $n \ge 1$:

$$
\begin{aligned}
a &\le c_1 n^2 \quad \rightarrow \quad c_1 = a \\
bn &\le c_2 n^2 \quad \rightarrow \quad c_2 = b
\end{aligned}
$$

$c = c_1 + c_2 = a + b$, and $n_0 = 1$

Approach 1: Focus on “c”

Proof.

Let $a, b \in \mathbb{R}^+$. Let $c = a + b$ and let $n_0 = 1$ Let $n \in \mathbb{N}$ and assume $n \geq n_0$. We’ll prove that $a + bn \leq cn^2$.

Since $1 \leq n$, we know $1 \leq n^2$, and so (multiplying by a), $a \leq an^2$.

Since $1 \leq n$, we know (multiplying by bn) that $bn \leq bn^2$.

Adding the previous two inequalities, we have:

$$\begin{aligned} a + bn &\leq an^2 + bn^2 \\ &= (a+b)n^2 \\ &= cn^2 \end{aligned}$$

Approach 2: Focus on “n”

$$\begin{aligned} a &\le c_1 n^2 \\ bn &\le c_2 n^2 \end{aligned}$$

Set $c_1 = c_2 = \frac{1}{2}$, and find n to satisfy:

$$\begin{aligned} a &\le \frac{1}{2} n^2 \\ bn &\le \frac{1}{2} n^2 \end{aligned}$$

$$\begin{aligned} a &\le \frac{1}{2} n^2 \quad \rightarrow \quad n \ge \sqrt{2a} \\ bn &\le \frac{1}{2} n^2 \quad \rightarrow \quad n \ge 2b \end{aligned}$$

Approach 2: Focus on n

$$a \leq \frac{1}{2}n^2 \quad \rightarrow \quad n \geq \sqrt{2a}$$
$$bn \leq \frac{1}{2}n^2 \quad \rightarrow \quad n \geq 2b$$

Pick n_0 so that $n \geq n_0$ implies $n \geq \sqrt{2a}$ and $n \geq 2b$.

$$\boxed{c = c_1 + c_2 = 1, \text{ and } n_0 = \max(\sqrt{2a}, 2b)}$$

Exercise 1: Practice with Big-O

Omega and Theta

Big-O expresses an upper bound on function growth.

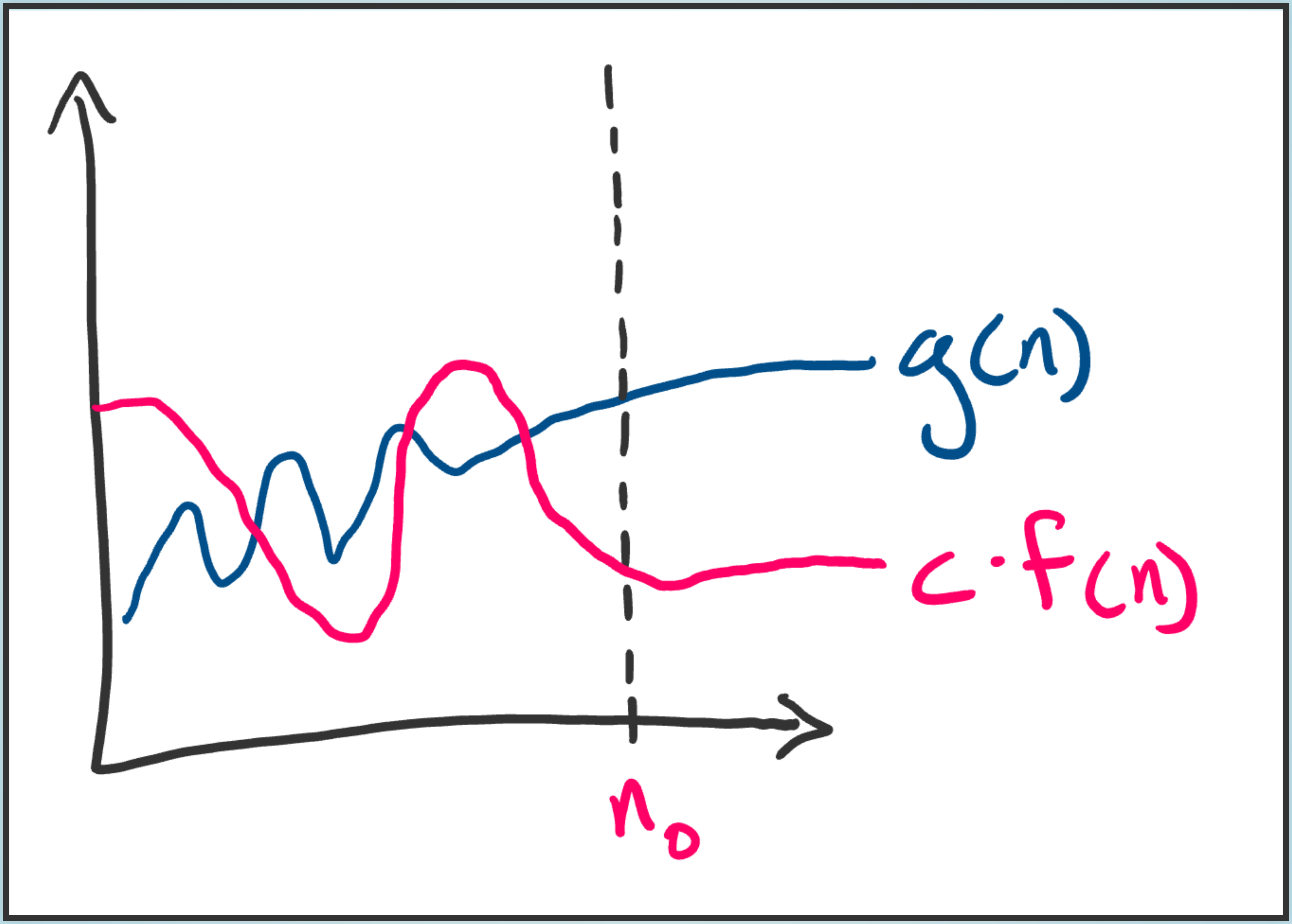
But these upper bounds might be very inaccurate!

$$10n + 5 \in \mathcal{O}(n^{1000})$$

Omega ("lower bound")

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. We say g **is Omega of** f and write $g \in \Omega(f)$ when:

$$\exists c, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow g(n) \geq c \cdot f(n)$$


g(n)
c·f(n)
n0

Proving "$g \in \Omega(f)$" is very similar to Big-O.

Proof.
Let $c = \ldots$ and $n_0 = \ldots$ Let $n \in \mathbb{N}$ and assume $n \geq n_0$.
We will prove that $g(n) \geq c \cdot f(n)$.

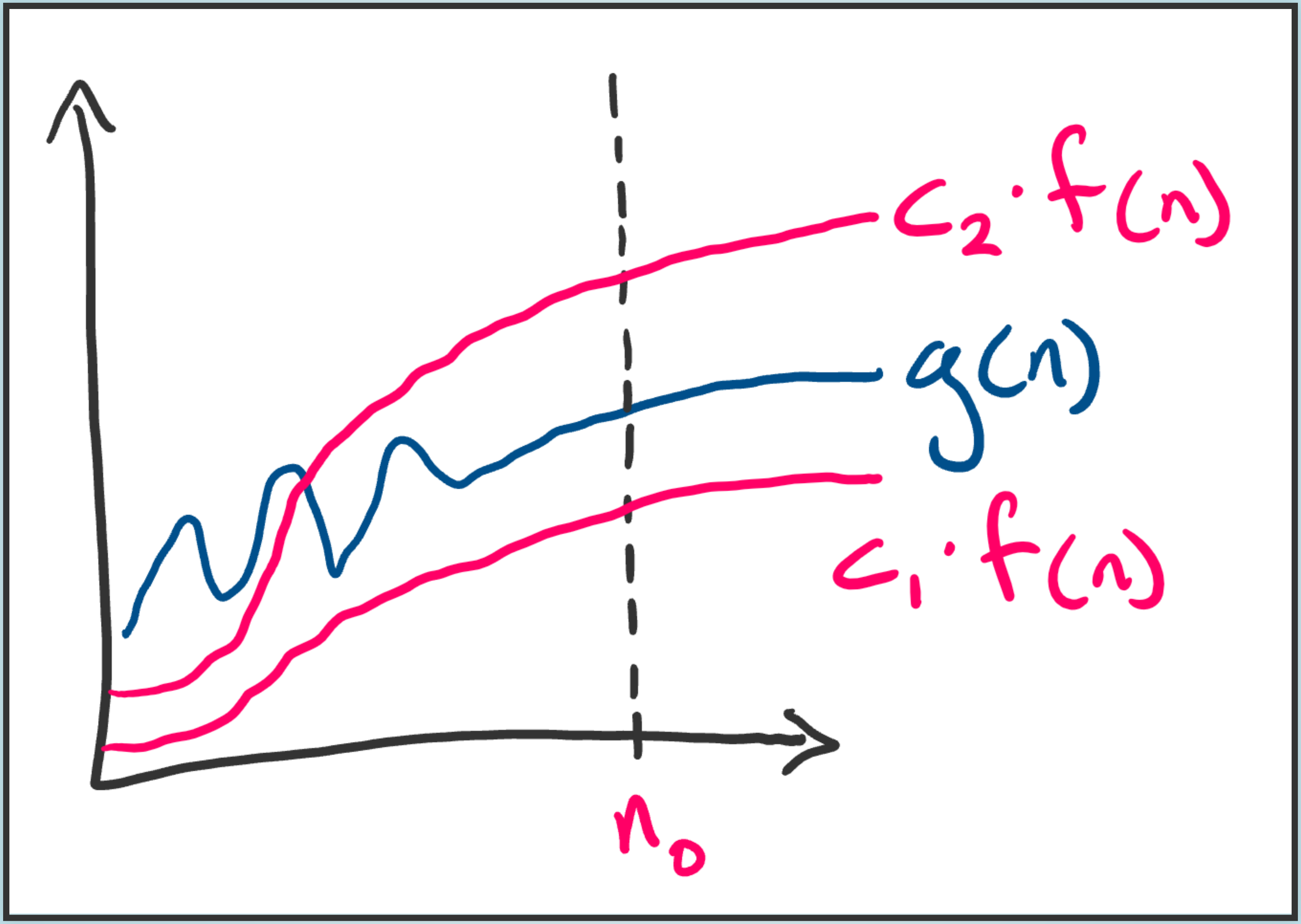
Theta

Let $f, g : \mathbb{N} \to \mathbb{R}^{\geq 0}$. We say g **is Theta of** f and write $g \in \Theta(f)$ when:

$$\exists c_1, c_2, n_0 \in \mathbb{R}^+,\ \forall n \in \mathbb{N},\ n \geq n_0 \Rightarrow c_1 \cdot f(n) \leq g(n) \leq c_2 \cdot f(n)$$

Or equivalently, when $g \in \mathcal{O}(f)$ and $g \in \Omega(f)$.

When $g \in \Theta(f)$ we say that f is a **tight bound** on g. (f is both an upper and lower bound on g)


$c_2 \cdot f(n)$
$g(n)$
$c_1 \cdot f(n)$
n_0

Proving "$g \in \Theta(f)$" involves proving two inequalities.

Proof.
Let $c_1 = \ldots$, $c_2 = \ldots$ and $n_0 = \ldots$ Let $n \in \mathbb{N}$ and assume $n \geq n_0$. We will prove that $g(n) \geq c_1 \cdot f(n)$ and $g(n) \leq c_2 \cdot f(n)$.

Big-O vs. Theta and tight bounds

Warning: when people say Big-O, they often mean Theta!

E.g., "$10 + 2n \in \mathcal{O}(n)$"

Definitions

Given $g \in \mathcal{O}(f)$, we say f is a **tight upper bound** on g when $g \in \Theta(f)$.

Given $g \in \Omega(f)$, we say f is a **tight lower bound** on g when $g \in \Theta(f)$.

Exercise 2: Omega and Theta

Comparing Elementary Functions

Powers of n

In Exercise 1, you proved that for all $a, b \in \mathbb{R}^+$, if $a < b$ then $n^a \in \mathcal{O}(n^b)$ and $n^b \notin \mathcal{O}(n^a)$.

What about other elementary functions?

Elementary Function Growth Hierarchy Theorem

For all $a, b \in \mathbb{R}^+$, the following statements are true:

1. If $a > 1$ and $b > 1$, then $\log_a n \in \Theta(\log_b n)$.
   - E.g., $\log_2 n \in \Theta(\log_{100} n)$
2. If $a < b$, then $n^a \in \mathcal{O}(n^b)$ and $n^a \notin \Omega(n^b)$.
   - E.g., $n^2 \in \mathcal{O}(n^{100})$ and $n^2 \notin \Omega(n^{100})$
3. If $a < b$, then $a^n \in \mathcal{O}(b^n)$ and $a^n \notin \Omega(b^n)$.
   - E.g., $2^n \in \mathcal{O}(100^n)$ and $2^n \notin \Omega(100^n)$

Elementary Function Growth Hierarchy Theorem, continued

4. If $a > 1$, then $1 \in \mathcal{O}(\log_a n)$ and $1 \notin \Omega(\log_a n)$.
   - Note: 1 means the constant function $g(n) = 1$ for all $n \in \mathbb{N}$

5. If $a > 1$, then $\log_a n \in \mathcal{O}(n^b)$ and $\log_a n \notin \Omega(n^b)$.
   - E.g., $\log_2 n \in \mathcal{O}(n^{0.0000000001})$ and $\log_2 n \notin \Omega(n^{0.0000000001})$

6. If $b > 1$, then $n^a \in \mathcal{O}(b^n)$ and $n^a \notin \Omega(b^n)$.
   - E.g., $n^{10000} \in \mathcal{O}(1.0000001^n)$ and $n^{10000} \notin \Omega(1.0000001^n)$

Summary

Today you learned to...

1. Define and explain the differences between Big-O, Omega, and Theta asymptotic notation.
2. Prove statements involving asymptotic notation.
3. Compare different elementary functions using asymptotic notation.

Homework

- Readings:
  - From prep: 9.1, 9.2
  - Today: 9.3
  - Next class: 9.3, 9.5
- Finish Assignment 3

Students when David mentions reading week