Problem set

Please turn in the problem set at the start of class.

COSC 290 Discrete Structures

Lecture 20: Asymptotics

Prof. Michael Hay

Monday, Oct. 16, 2017

Colgate University

Plan for today

1. Big-Oh definition
2. Proving $f(n) = O(g(n))$ or $f(n) \neq O(g(n))$
3. Properties of Big-Oh
4. Discuss mid-semester feedback

Big-Oh definition

Main use for big-Oh notation

$f(n)$ is typically the runtime of some algorithm $\mathcal{A}$ on an input of size n.

$g(n)$ is an upper bound on the runtime of the algorithm, ignoring constants.

Why ignore constants?

Big Oh Notation

Let $f : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ and $g : \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$.

We say that *f grows no faster than g* if there exists some $n_0 \in \mathbb{R}^{\geq 0}$ and some $c \in \mathbb{R}^{>0}$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

This is denoted $f(n) = O(g(n))$.

A graphical view

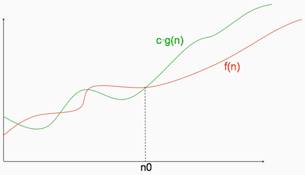

Figure 1: $f(n) = O(g(n))$. For $n \geq n_0$, $f(n)$ grows no faster than $c \cdot g(n)$.

Poll: formalized logically

We say $f(n) = O(g(n))$ when there exists some $n_0 \in \mathbb{R}^{\geq 0}$ and some $c \in \mathbb{R}^{>0}$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

How do we formalize this as a proposition?

A) $\forall n \in \mathbb{R}^{\geq 0} : \exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : (n \geq n_0) \wedge (f(n) \leq c \cdot g(n))$

B) $\forall n \in \mathbb{R}^{\geq 0} : \exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : (n \geq n_0) \implies (f(n) \leq c \cdot g(n))$

C) $\exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \wedge (f(n) \leq c \cdot g(n))$

D) $\exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \implies (f(n) \leq c \cdot g(n))$

E) More than one / None of the above

Proving $f(n) = O(g(n))$ or $f(n) \neq O(g(n))$

Proving a big-Oh relationship

We say $f(n) = O(g(n))$ when there exists some $n_0 \in \mathbb{R}^{\geq 0}$ and some $c \in \mathbb{R}^{>0}$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

$$\exists n_0 \in \mathbb{R}^{\geq 0} : \exists c \in \mathbb{R}^{>0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \implies (f(n) \leq c \cdot g(n))$$

To prove $f(n) = O(g(n))$, ...

- you get to *choose* values for n_0 and c.
- Then, start with $f(n)$ and...
- ... use inequalities to show it's no larger than $c \cdot g(n)$.

Problem set question

6.5. Let $f(n) = 9n + 3$ and let $g(n) = n$. Prove that $f(n) = O(g(n))$.

Choose values for n_0 and c:

- $n_0 = 3$ *(other choices work)*
- $c = 10$ *(other choices work)*

Want to show: for $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

$$\begin{aligned} f(n) &= 9n + 3 \\ &\leq 9n + n && \text{when } n \geq n_0 \\ &= 10n \\ &\leq c \cdot g(n) && \text{because } c = 10. \ \square \end{aligned}$$

Problem set question

6.7. Let $f(n) = 9n + 3$ and let $g(n) = 3n^3 - n^2$. Prove that $f(n) = O(g(n))$.

Choose $n_0 = 3$ and $c = 5$ *(other choices also work)*.

Want to show: for $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

$$\begin{aligned} f(n) = 9n + 3 &\leq 9n + n && \text{when } n \geq n_0 \\ &= 10n \\ &\leq 10n^3 && \text{since } n^2 \geq 1 \text{ when } n \geq n_0 \\ &= 5 \cdot 2n^3 = 5 \cdot (3n^3 - n^3) && \text{algebra} \\ &\leq 5 \cdot (3n^3 - n^2) && \text{because } n^2 \leq n^3 \\ &= c \cdot g(n) && \square \end{aligned}$$

Exercise

Let $f(n) = 10^6 n^2 + 10^3 n$ and let $g(n) = n^2$. Show that $f(n) = O(n^2)$.

Work in small groups. Raise your hand when you have an answer.

Poll: formalized logically

We say $f(n) = O(g(n))$ when

$$\exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \implies (f(n) \leq c \cdot g(n))$$

How do we formalize $f(n) \neq O(g(n))$ as a proposition?

A) $\exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : \forall n \in \mathbb{R}^{\geq 0} : \neg(n \geq n_0) \implies \neg(f(n) \leq c \cdot g(n))$
B) $\forall c \in \mathbb{R}^{>0} : \forall n_0 \in \mathbb{R}^{\geq 0} : \exists n \in \mathbb{R}^{\geq 0} : \neg(n \geq n_0) \implies \neg(f(n) \leq c \cdot g(n))$
C) $\exists c \in \mathbb{R}^{>0} : \exists n_0 \in \mathbb{R}^{\geq 0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \wedge (f(n) > c \cdot g(n))$
D) $\forall c \in \mathbb{R}^{>0} : \forall n_0 \in \mathbb{R}^{\geq 0} : \exists n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \wedge (f(n) > c \cdot g(n))$
E) More than one / None of the above

Proving a big-Oh relationship does *not* hold

We say $f(n) \neq O(g(n))$ when for all $n_0 \in \mathbb{R}^{\geq 0}$ and all $c \in \mathbb{R}^{>0}$ there exists $n \geq n_0$ such that $f(n) > c \cdot g(n)$.

$$\forall c \in \mathbb{R}^{>0} : \forall n_0 \in \mathbb{R}^{\geq 0} : \exists n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \wedge (f(n) > c \cdot g(n))$$

To prove $f(n) \neq O(g(n))$, ...

- you *cannot* choose values for n_0 and c, so let them be arbitrary values.
- You do get to choose any $n \geq n_0$.
- Then, start with $f(n)$ and...
- ... use inequalities to show it's larger than $c \cdot g(n)$.

Problem set question

6.10. Let $g(n) = 3n^3 - n^2$ and $h(n) = n^2$. Prove that $g(n) \neq O(h(n))$.

Let $n_0 \in \mathbb{R}^{\geq 0}$ and $c \in \mathbb{R}^{>0}$ be arbitrary.

Want to show: there is some $n \geq n_0$ where $g(n) > c \cdot h(n)$.

Let $n := \max\{n_0, c + 1\}$ so that $n \geq n_0$ and $n > c$:

$$\begin{aligned} g(n) &= 3n^3 - n^2 \\ &\geq 3n^3 - n^3 && n^3 \geq n^2 \text{ because } n \geq 1 \\ &= 2n^3 \\ &> cn^2 && \text{because } n > c \\ &= c \cdot g(n) && \square \end{aligned}$$

Properties of Big-Oh

Useful properties

- If $f(n) = \log n$, $f(n) = O(n)$.
- If $f(n)$ is a degree k polynomial, $f(n) = O(n^k)$.
- If $f(n) = n^k$, $f(n) = O(2^n)$.
- If $f(n) = O(g(n))$ and $g(n) = O(h(n))$, then $f(n) = O(h(n))$.
- If $f(n) = O(h_1(n))$ and $g(n) = O(h_2(n))$, then
  - $f(n) + g(n) = O(h_1(n) + h_2(n))$, and
  - $f(n) \cdot g(n) = O(h_1(n) \cdot h_2(n))$.
- $f(n) = O(g(n) + h(n)) \iff f(n) = O(\max(g(n), h(n)))$

Exercise

Recall definition of big-Oh: we say $f(n) = O(g(n))$ when there exists some $n_0 \in \mathbb{R}^{\geq 0}$ and some $c \in \mathbb{R}^{>0}$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.

$$\exists n_0 \in \mathbb{R}^{\geq 0} : \exists c \in \mathbb{R}^{>0} : \forall n \in \mathbb{R}^{\geq 0} : (n \geq n_0) \implies (f(n) \leq c \cdot g(n))$$

Exercise: Use this to prove that if $f(n) = O(h_1(n))$ and $g(n) = O(h_2(n))$, then $f(n) \cdot g(n) = O(h_1(n) \cdot h_2(n))$.

Work in small groups. Raise your hand when you have an answer.

Big Theta and the rest

1. $f(n) = O(g(n))$ says f grows no faster than $g(n)$
2. $f(n) = \Omega(g(n))$ says f grows no slower than $g(n)$
3. $f(n) = \Theta(g(n))$ says f grows at the same rate as $g(n)$
4. $f(n) = o(g(n))$ says f grows (strictly) slower faster than $g(n)$
5. $f(n) = \omega(g(n))$ says f (strictly) faster than $g(n)$

See text for formal definitions. You are expected to become familiar with this notation, especially first three!

Discuss mid-semester feedback

Some comments

- Course is challenging for some but "well paced" for others
- Almost all valued in-class questions and discussion
- Less frequent comments I want to address
  - Asking questions in class
  - More "individualized attention"
  - More time to copy materials on slides
  - Grading fairness – almost all agreed grading was fair but a few did not (please come see me!)