Asymptotic Analysis

CPSC 320 2024S

Asymptotic bounds

- It's useful to be able to compare the growth rate of running times in a way that's insensitive to constant factors and lower-order terms. For this we use O, Ω, and Θ.
- We say that $f \in O(g)$ if there exist positive constants c and n_0 such that, for all $n \geq n_0$, we have $f(n) \leq c \cdot g(n)$.
  - Example: $2n \in O(n)$ because for any $c \geq 2$, we have $2n \leq cn$ for all positive n. We also have $2n \in O(n^2)$, because if we choose (for example) $c = 1$, then $2n \leq cn^2$ for all $n \geq 2$.
- Conversely, $f \in \Omega(g)$ exactly when $g \in O(f)$.
- $f \in \Theta(g)$ (and $g \in \Theta(f)$) when $f \in \Omega(g)$ and $f \in O(g)$.

Not exactly “best” and “worst” cases

- Common misunderstanding: “big-O means the worst case and big-Omega means the best case”
- Best-case and worst-case are properties of an algorithm.
  - E.g., worst-case for QuickSort with first element chosen as pivot is a list that’s already sorted (gives a $\Theta(n^2)$ runtime).
- O and Ω are ways to compare functions and don’t *inherently* have anything to do with algorithms.
  - E.g., if $f(n) = 3n + 16$ then $f(n) \in O(n)$ and $f(n) \in O(n^2)$.

Clicker Question #1

Suppose I told you to give an $\boldsymbol{\Omega}$**-bound** on the **worst-case runtime** of QuickSort. Which of the following bounds apply?

A. $\Omega(n \log n)$

B. $\Omega(n^2)$

C. Both A and B

D. Neither A nor B

E. It is impossible to provide an Ω-bound on the worst-case runtime of a function

Clicker Question #1

Suppose I told you to give an ***Ω*-bound** on the **worst-case runtime** of QuickSort. Which of the following bounds apply?

A. $\Omega(n \log n)$

B. $\Omega(n^2)$

C. Both A and B

D. Neither A nor B

E. It is impossible to provide an Ω-bound on the worst-case runtime of a function

Little-o/ω

- Big O, Ω, and Θ are *roughly* the asymptotic equivalents of $\leq, =, \geq$ on functions. Little-o and little-ω are like $<$ and $>$.
- A function f is little-o of another function g if the growth of f is *strictly slower* than g
- Formally: $f \in o(g)$ when, for **every** positive real number c, there is a positive integer n_0 such that, for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.
  - Example: $2n \in o(n^2)$ because even for very small c, we will have $2n \leq cn^2$ for all sufficiently large n. However, $2n \notin o(n)$ because we do not have $2n \leq cn$ for, say, $c = 0.1$.
- Conversely, $f \in \omega(g)$ exactly when $g \in o(f)$

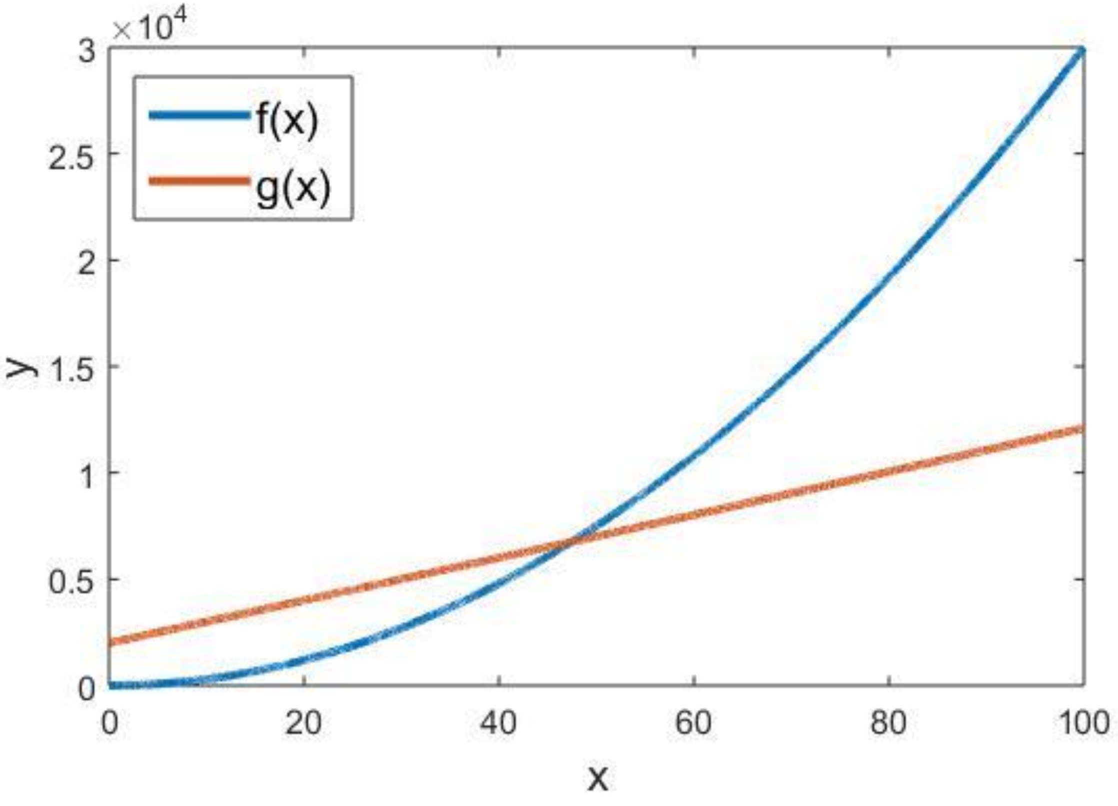
Little-o/ω definitions via limits

- A handy tool to observe how functions are related asymptotically is to compare the value of $f(n)/g(n)$ as n gets very large
- When $\lim_{n\to\infty} \frac{f(n)}{g(n)}$ is well-defined, we can apply the following:
  - If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$ then $g(n) \in o\big(f(n)\big)$ and $f(n) \in \omega\big(g(n)\big)$
  - If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$ then $f(n) \in o\big(g(n)\big)$ and $g(n) \in \omega\big(f(n)\big)$
  - If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$ (not equal to ∞ or 0) then $f(n) \in \Theta(g(n))$ (and $g(n) \in \Theta(f(n))$)

Clicker Question #2

What is the relationship between $f(x)$ and $g(x)$?

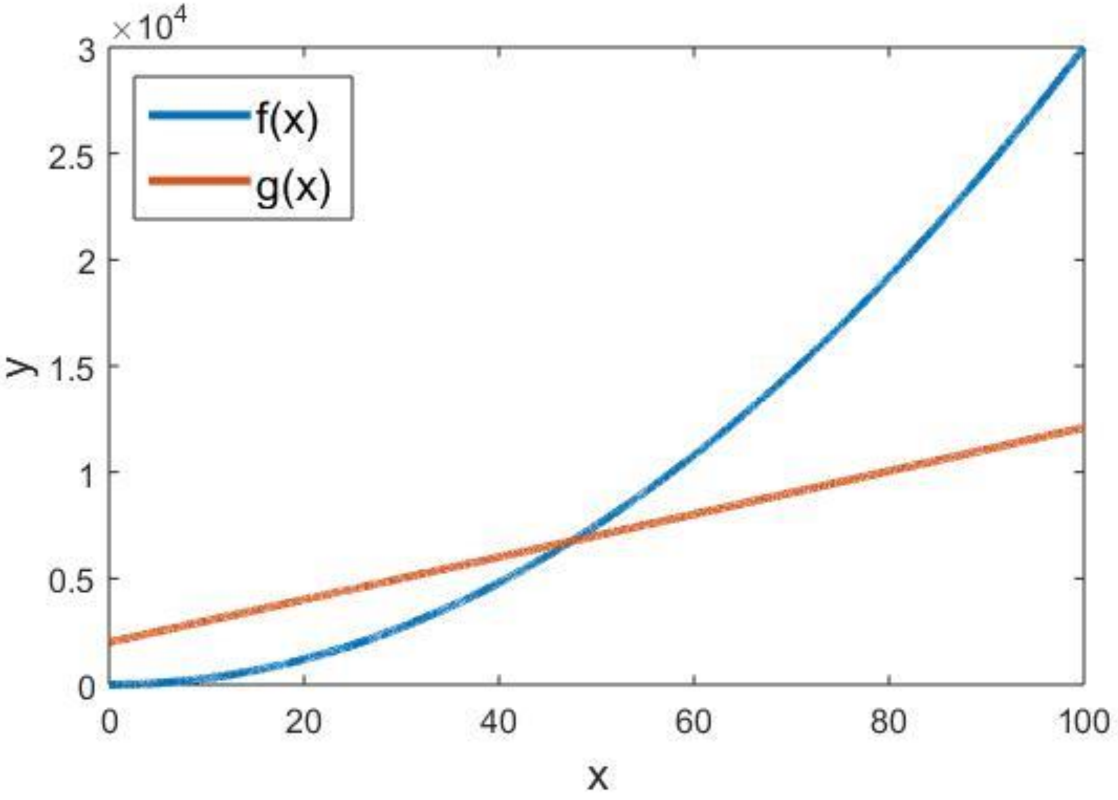
A. $f(x) \in o\big(g(x)\big)$
B. $f(x) \in \Theta\big(g(x)\big)$
C. $f(x) \in \omega(g(x))$
D. None of the above relations hold for all x
E. Impossible to determine



Clicker Question #2

What is the relationship between $f(x)$ and $g(x)$?

A. $f(x) \in o\big(g(x)\big)$
B. $f(x) \in \Theta\big(g(x)\big)$
C. $f(x) \in \omega(g(x))$
D. None of the above relations hold for all x
E. Impossible to determine



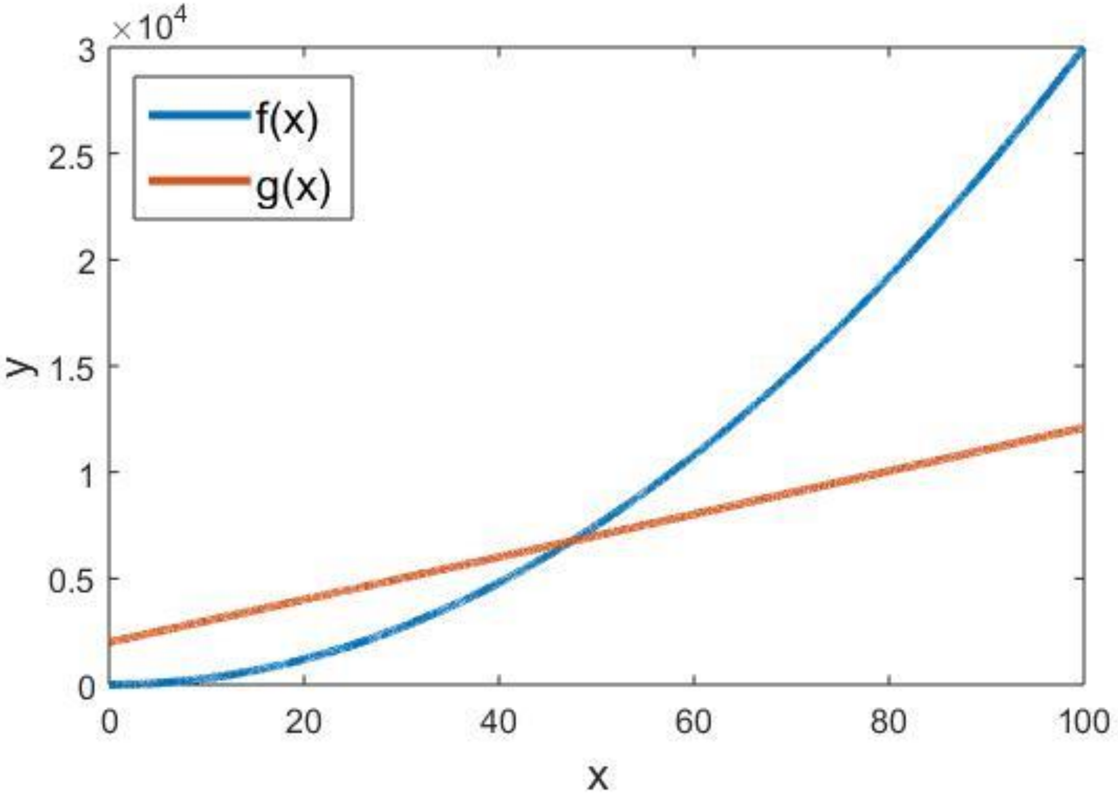
Clicker Question #2

$f(x) = 3x^2$

$g(x) = \frac{1}{1000}x^3 + 100x + 2000$

What is the relationship between $f(x)$ and $g(x)$?

A. $f(x) \in o\big(g(x)\big)$
B. $f(x) \in \Theta\big(g(x)\big)$
C. $f(x) \in \omega(g(x))$
D. None of the above relations hold for all x
E. Impossible to determine



Clicker Question #3

Which is the **worst** (i.e., fastest-growing) of the following runtimes?

A. $\log(n^{100})$

B. $(\log n)^{100}$

C. $\sqrt[100]{n}$

Clicker Question #3

Which is the **worst** (i.e., fastest-growing) of the following runtimes?

A. $\log(n^{100})$

B. $(\log n)^{100}$

C. $\sqrt[100]{n}$