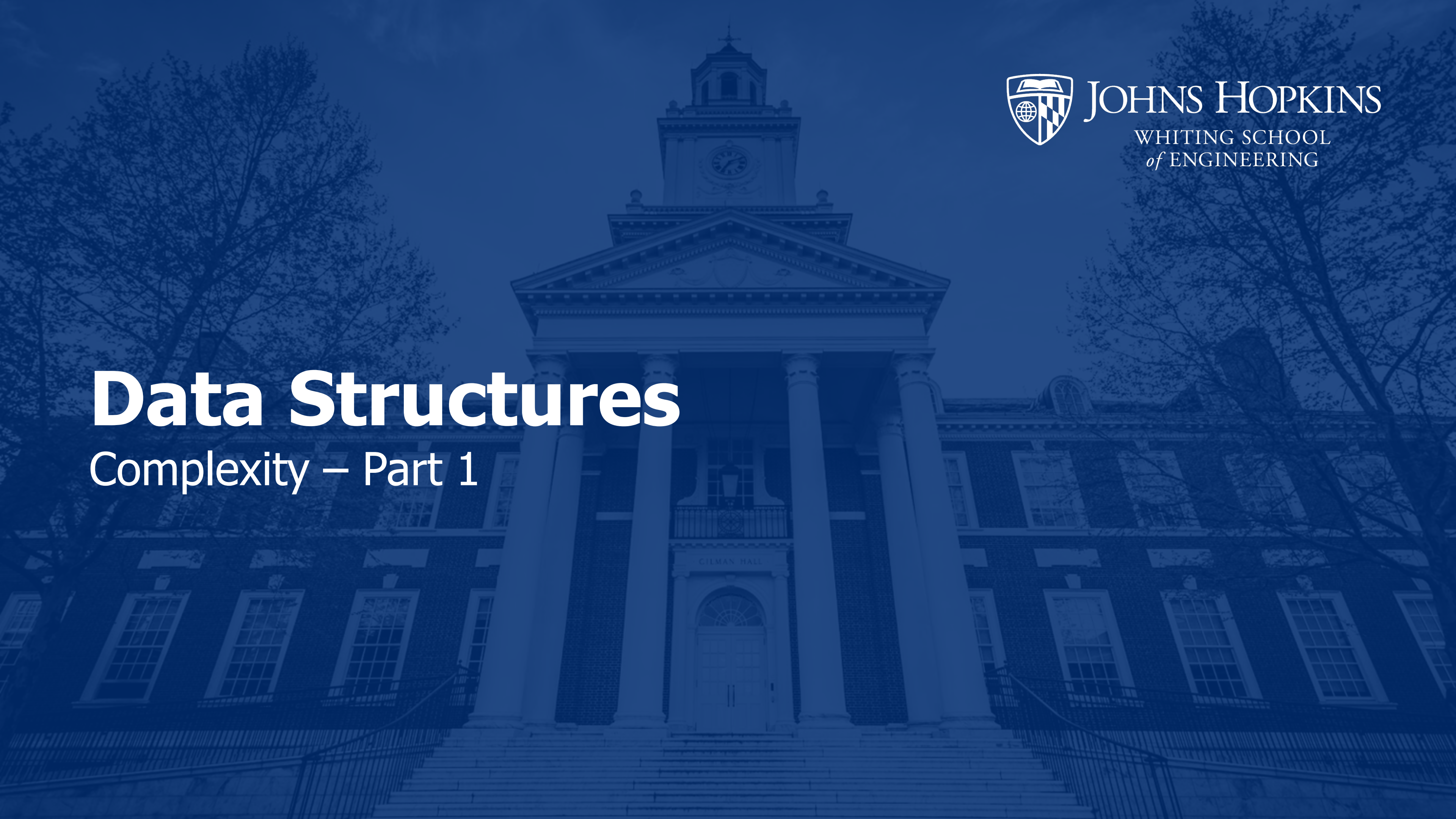


JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Data Structures

Complexity – Part 1

Complexity

- People naturally make comparisons
  - e.g., Car A is better than Car B!
- How do we define “better”?

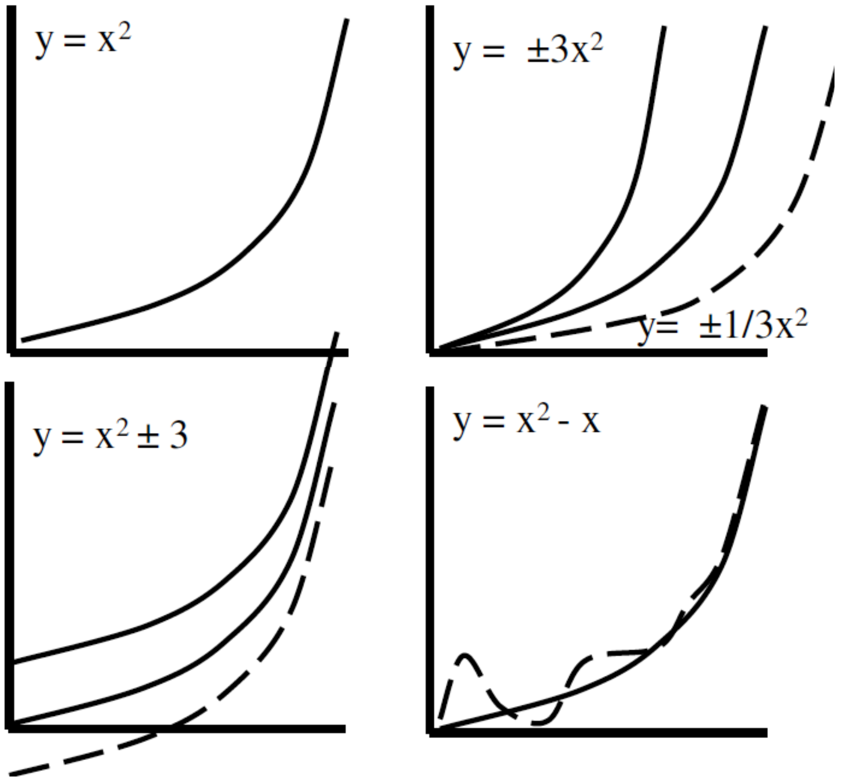
  A. A has more horsepower than

  B. B holds 7 people, but A only holds 2
- Each can be better than the other in a specific, useful context.

Can We Do This With Programs?

- YES, we call this “Big **O**” Notation.
- Label code to allow comparisons What does ***better*** mean here?

Review of Simple Planar Functions



A Review of Simple, Planar Functions:

- What decides the shape of the curve?
- This is independent of:
  - the lower order terms
  - the coefficients used

Deriving Work Done By Code (1)

Given a specific language,

a specific operating system and

a specific compiler:

Consider this assignment:

$$x = x + 1$$

How much does it take?

Deriving Work Done By Code (2)

- Suppose we change the system?
- Suppose:

  $\text{for } (int\ i = 1; i{<=}n; i++)$

  $x = x + 1;$
- Suppose:

  $\text{for } (int\ i = 1; i{<=}n; i++)$

  $\text{for } (int\ j = 1; j{<=}n; j++)$

  $x = x + 1;$

Deriving Work Done By Code (3)

For any piece of code, generate a function to represent the work done:

For example:

$$f(n) = c_1 n + c_2 + c_3 n + c_4 n^3 + c_5 n^2 + c_6 + c_7 n^2 + c_8$$

Simplifying:

$$f(n) = c_4 n^3 + (c_5 + c_7) n^2 + (c_1 + c_3) n + (c_2 + c_6 + c_8)$$

Deriving Work Done By Code (4)

- **This is messy to graph.**
- If all we are interested in is the basic shape, we can simplify
- by using the dominant term.
- This gives us a label to use for the code whose work is represented by f(x).

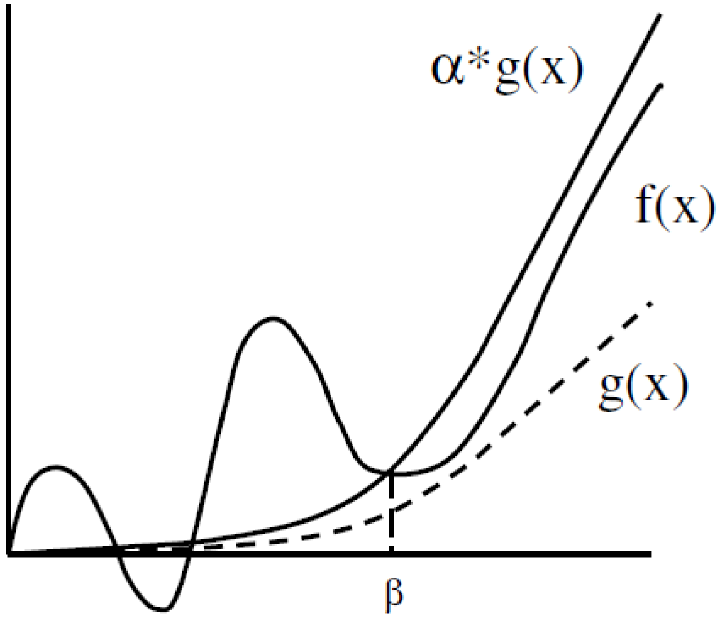
Definition: Upper Bound

- Give two functions $f(n)$ and $g(n)$ and two real constants $\propto$ and β;

  if $\alpha^{*}g(n) \geq f(n)$, for all $n > \beta$ then $g(n)$ is an upper bound for $f(n)$

  f is said to be $\mathrm{O(g(n))}$

Definition: Upper Bound (cont.)



Upper Bounds

- In particular, if f(n) is a polynomial then g(n) is the dominant term.
- g(n) is an estimate of how f(n) acts.
- We are guaranteed that f will do no worse than g. It might do better.

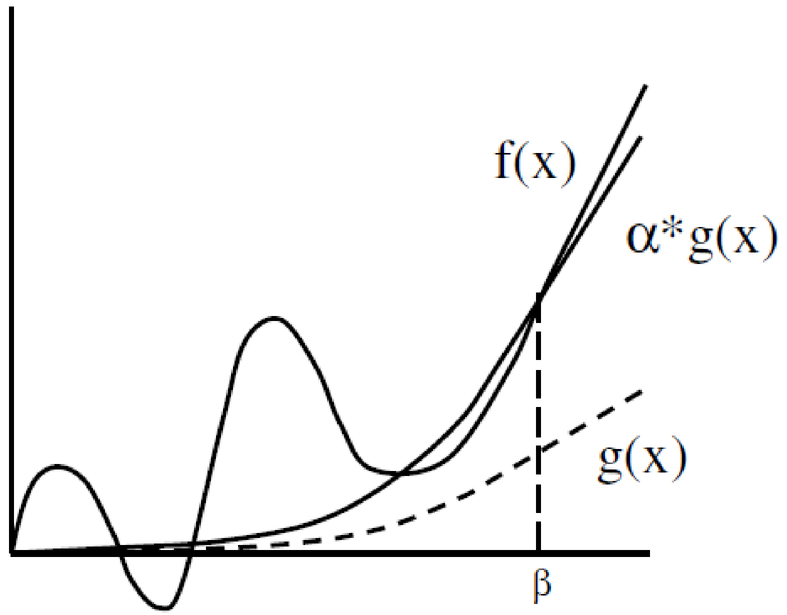
Definition: Lower Bound

- Give two functions $f(n)$ and $g(n)$ and two real constants $\propto$ and β;

  if $\alpha * g(n) \leq f(n)$, for all $n > \beta$ then $g(n)$ is an lower bound for $f(n)$

  f is said to be $\Omega(g(n))$

Definition: Lower Bound (cont.)



Lower Bounds

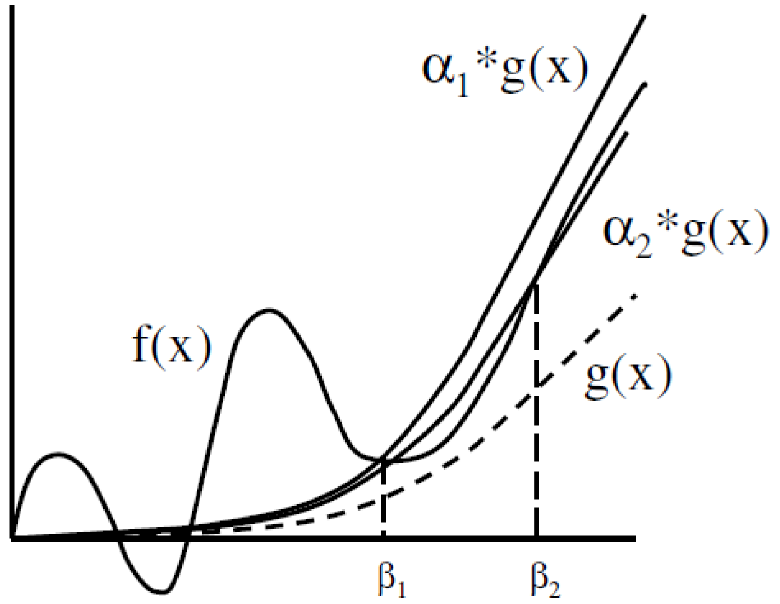
- In particular, if f(n) is a polynomial then g(n) is the dominant term.
- g(n) is an estimate of how f(n) acts.
- We are guaranteed that f is no better than g. It might be worse.

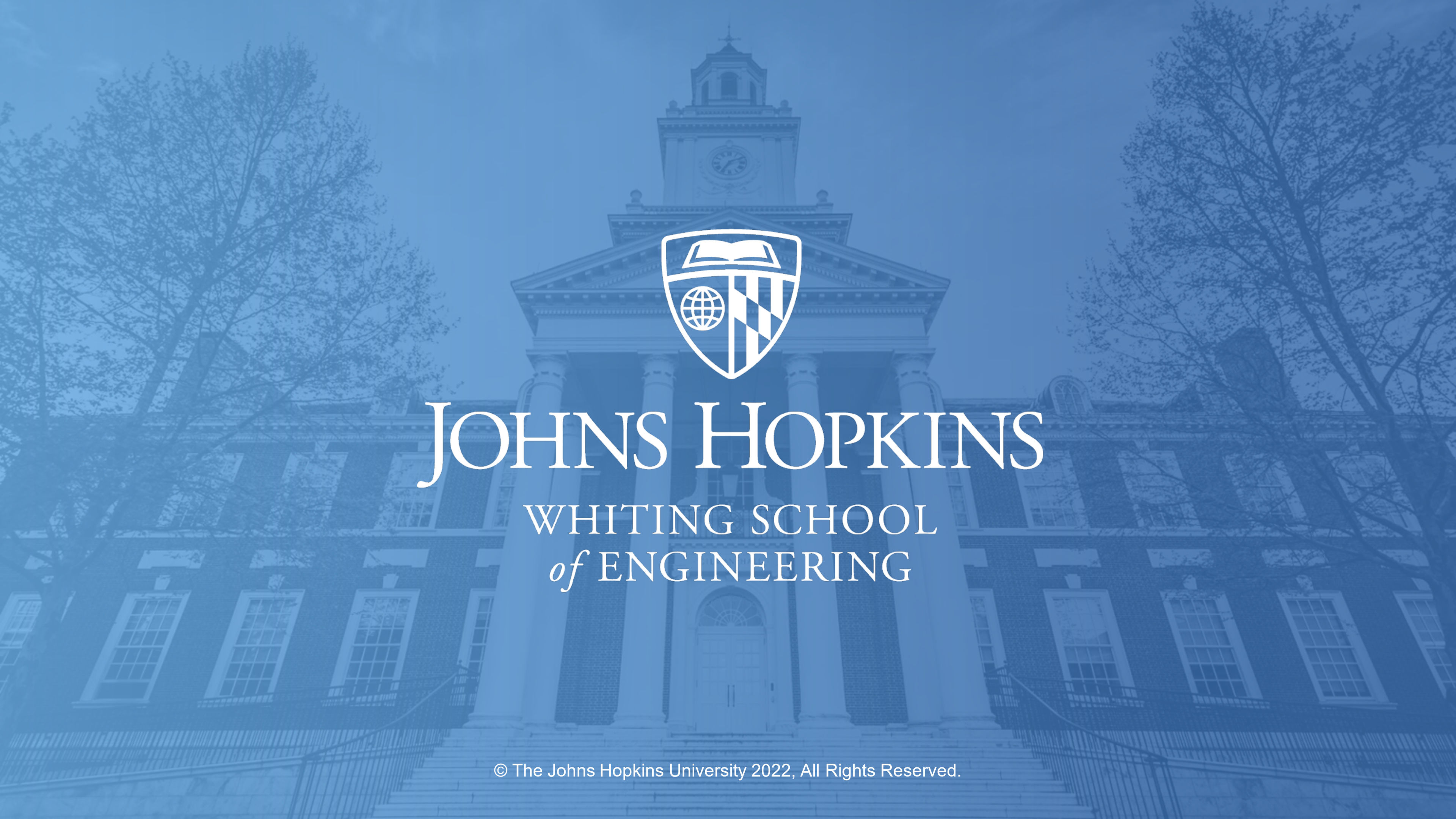
Both

- If $f(n)$ is $\mathrm{O}(\mathrm{g}(\mathrm{n}))$ and $\Omega(g(n))$ then f is said to be $\Theta(g(n))$
  g is an upper bound for f and g is a lower bound for f.

e.g. $\alpha_2^* g(n) \leq f(n) \leq \alpha_1^* g(n)$

Definition: Both Bounds





JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING