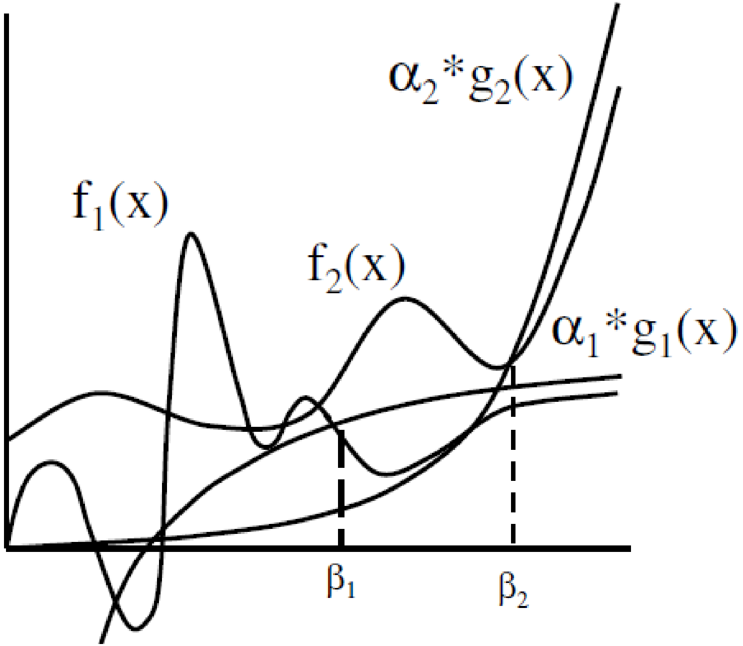
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Data Structures

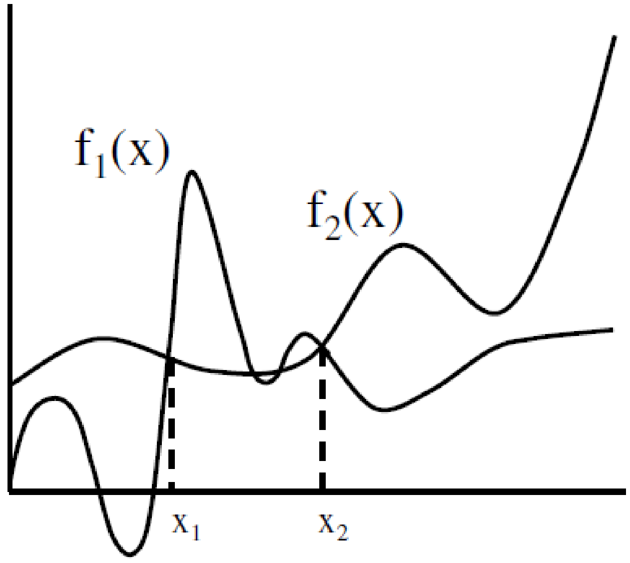
Complexity – Part 2

Example (1)



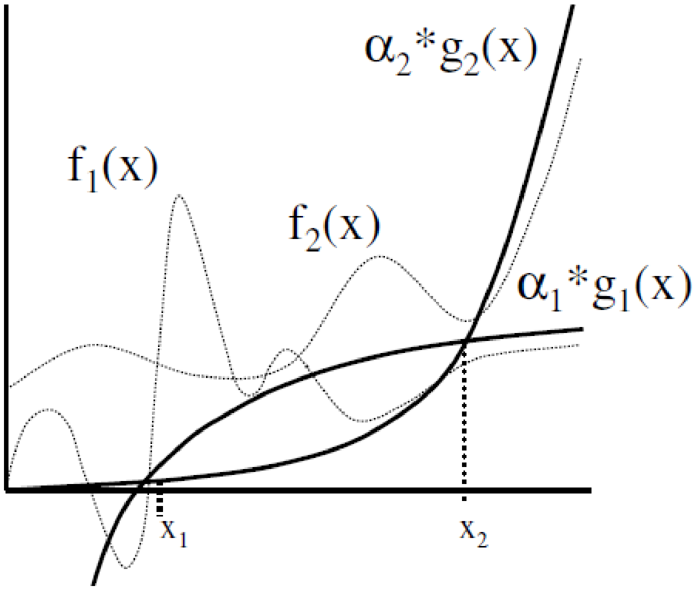
Example (2)

Which function should we pick?



Example (3)

Now which function should we pick?



Practical Issues

- Most books use “Big O”
- Sometimes “Big Theta”
- Sometimes "Big Omega"
- Usually, we look at the time required.
- Sometimes we look at the space required.
- Remember, g is only a bound beyond the specified point β

More Practical Issues

Traditionally

- log n implies base 10 lg n implies base 2
- ln n implies base e
- Other bases are specified logb n

More Practical Issues (cont.)

Sometimes, $\log n$ is used but $\log_2 n$ or $\lg n$ is implied by context

The identity $\log_b a = \dfrac{\log_c a}{\log_c b}$

makes the conversion a constant

Standard "Big O" Values

Polynomial time (P)

Nondeterministically Polynomial Time (NP)

Computer scientists believe $P \subseteq NP$

This is not proven.

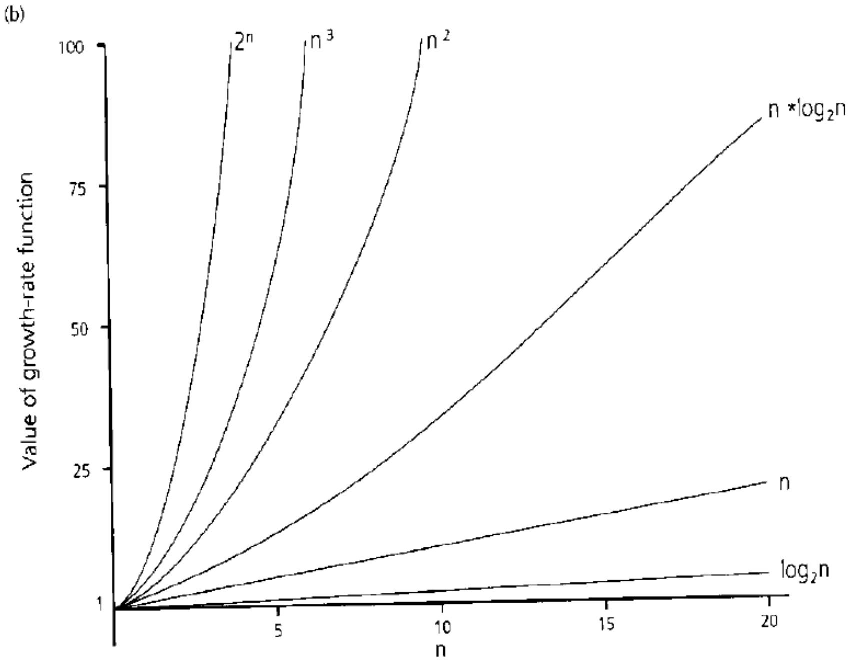
"Big O": Polynomial time (P)

$O(1)$	constant time
$O(\log n)$	log time
$O(n)$	linear time
$O(n \log n)$	
$O(n^2)$	quadratic time
$O(n^3)$	cubic time
$\vdots$	
$O(n^k)$	

Functions

Function	10	100	1,000	10,000	100,000	1,000,0001
1	1	1	1	1	1	1
$\log_2 n$	3	6	9	13	16	19
n	10	10^2	10^3	10^4	10^5	10^6
$n \log_2 n$	30	364	9965	10^5	10^6	10^7
n^2	10^2	10^4	10^6	10^8	10^{10}	10^{12}
n^3	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}
2^n	10^3	10^{30}	10^{301}	$10^{3,010}$	$10^{30,103}$	$10^{301,030}$

Value of Growth-Rate Function



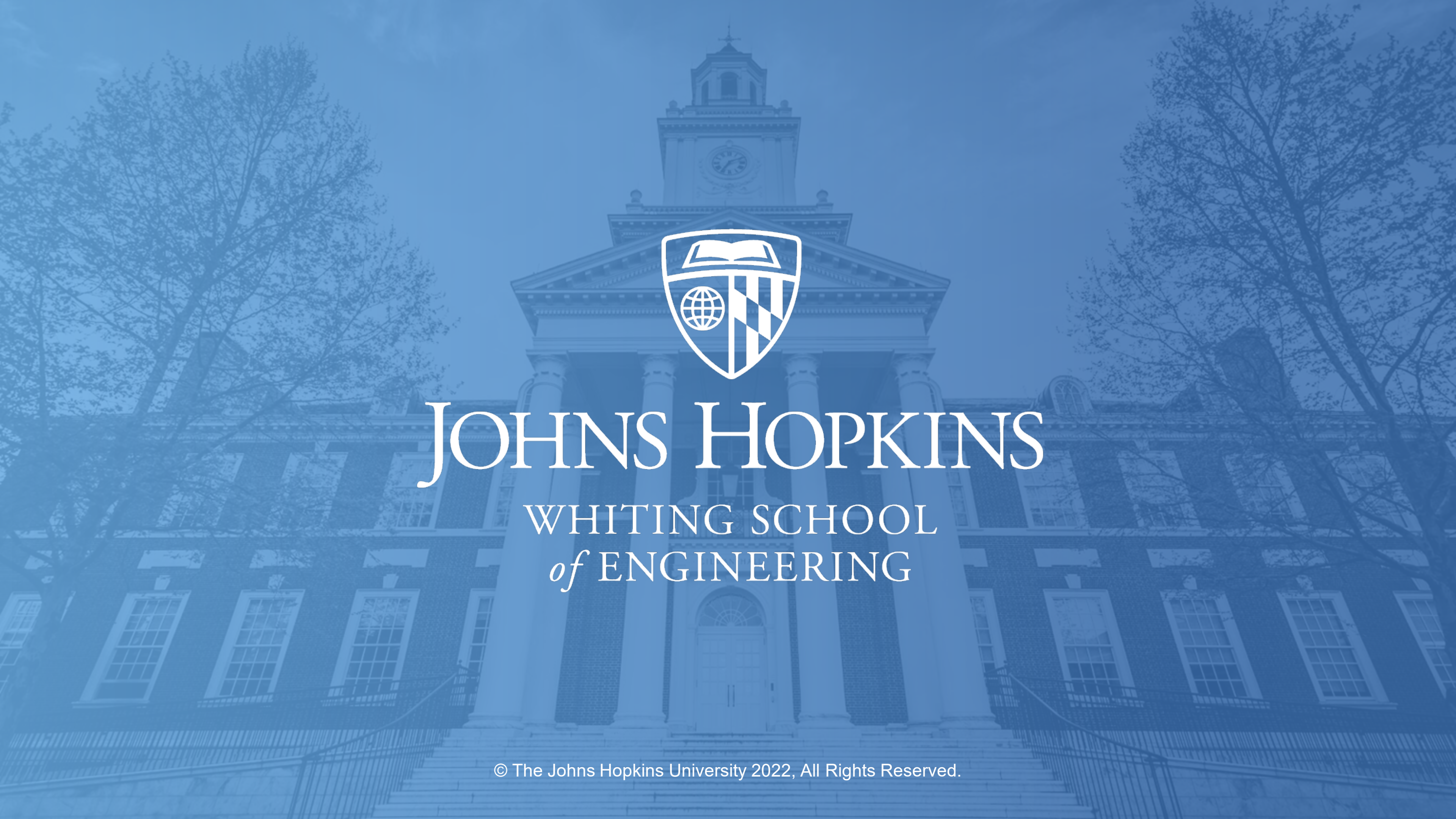
More Complexity Theory

- Tune in to 605.421 for more exciting complexity theory.
- In this course, algorithms range from constant time to cubic time,

  e.g. Sorts range from $O(n)$ to $O(n^2)$

Remember:

- O(g(n)) is an **estimate** of performance.
- Possibly:
  - The “worst” algorithm is sometimes the best choice



JOHNS HOPKINS

WHITING SCHOOL *of* ENGINEERING