COMP170
Discrete Mathematical Tools for Computer Science
Big O Notation

Version 2.1: Last updated, Nov 3, 2008

A quick and dirty Introduction to big O Notation

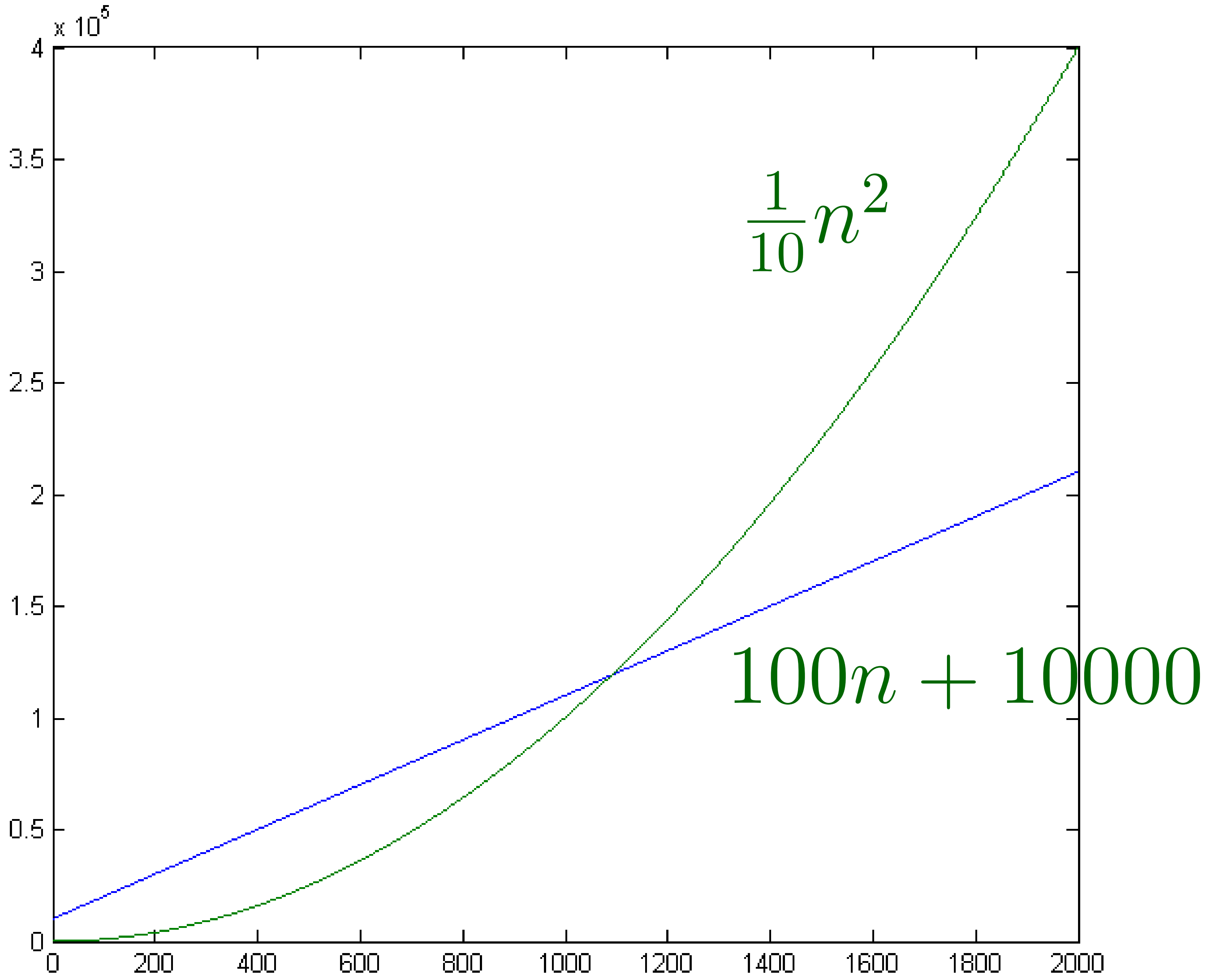
(You'll see more details in COMP171 and COMP271.)

Which function is "bigger"?

$\frac{1}{10}n^2$ or $100n + 10000$

Answer depends upon value of n.

In Computer Science we are usually interested in what happens when our problem input size gets large.



x 10^5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
200
400
600
800
1000
1200
1400
1600
1800
2000
$\frac{1}{10}n^2$
$100n + 10000$

Notice that when n is "large enough" $\frac{1}{10}n^2$ gets much bigger than $100n + 10000$ and stays larger.

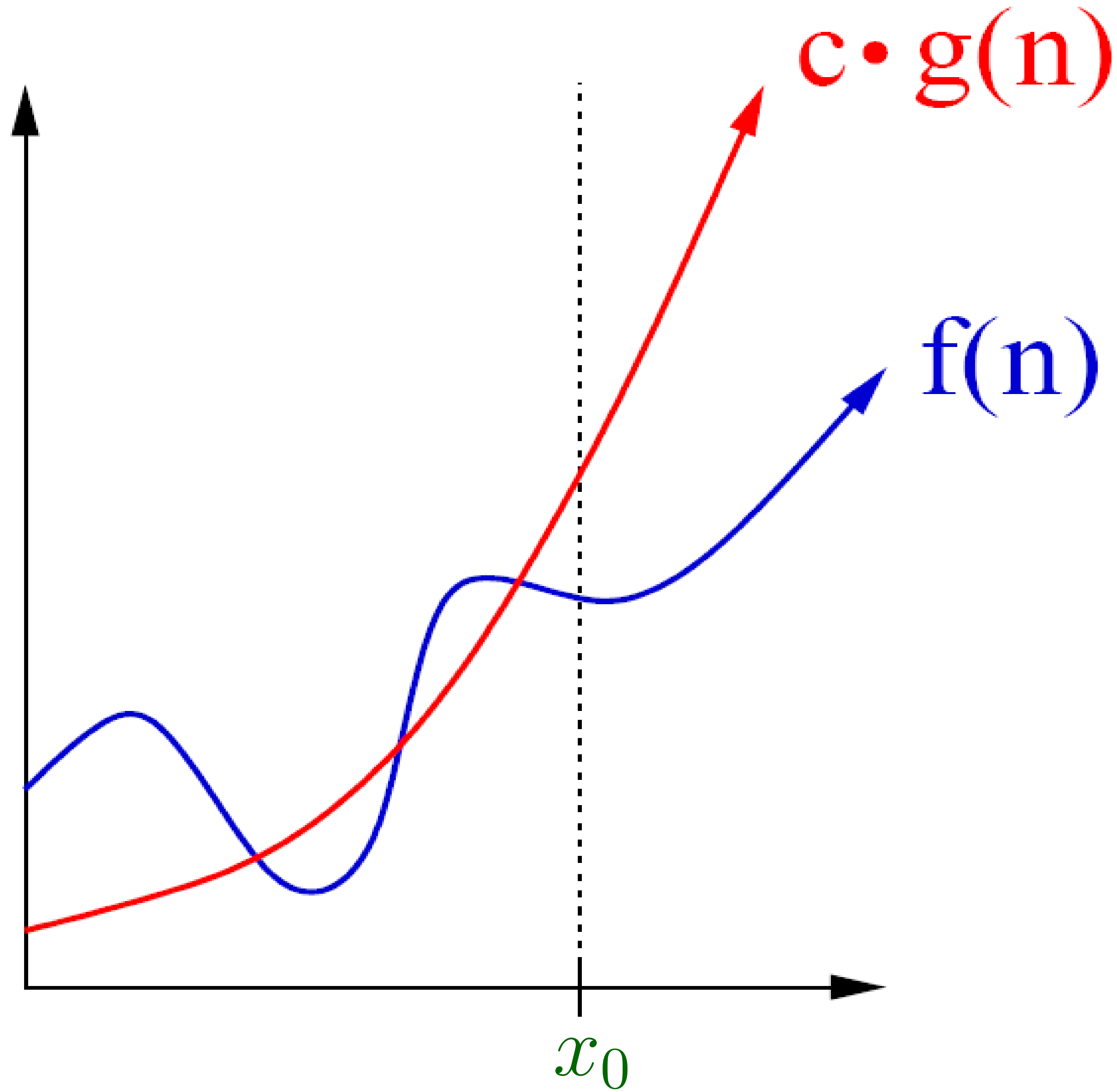
Function $f(n) = O(g(n))$:

(read: $f(n)$ is O of $g(n)$)

If (i) There is some positive $x_0 \in R$

(ii) There is some positive $c \in R$ such that

$$\forall x \geq x_0 \qquad f(x) \leq cg(x).$$


c • g(n)
f(n)
x_0

Let $x_0 = 1091$.
Can verify, $\forall n > x_0$, $100n + 10000 \le \frac{1}{10}n^2$.
Thus $\quad 100n + 10000 = O(\frac{1}{10}n^2)$.

Note that the opposite is **not** true!

Why? *(Proof by contradiction)*

More Examples:

$$4n^2$$
$$8n^2 + 2n - 3$$
$$n^2/5 + \sqrt{n} - 10\log n$$
$$n(n-3)$$

are all $O(n^2)$.

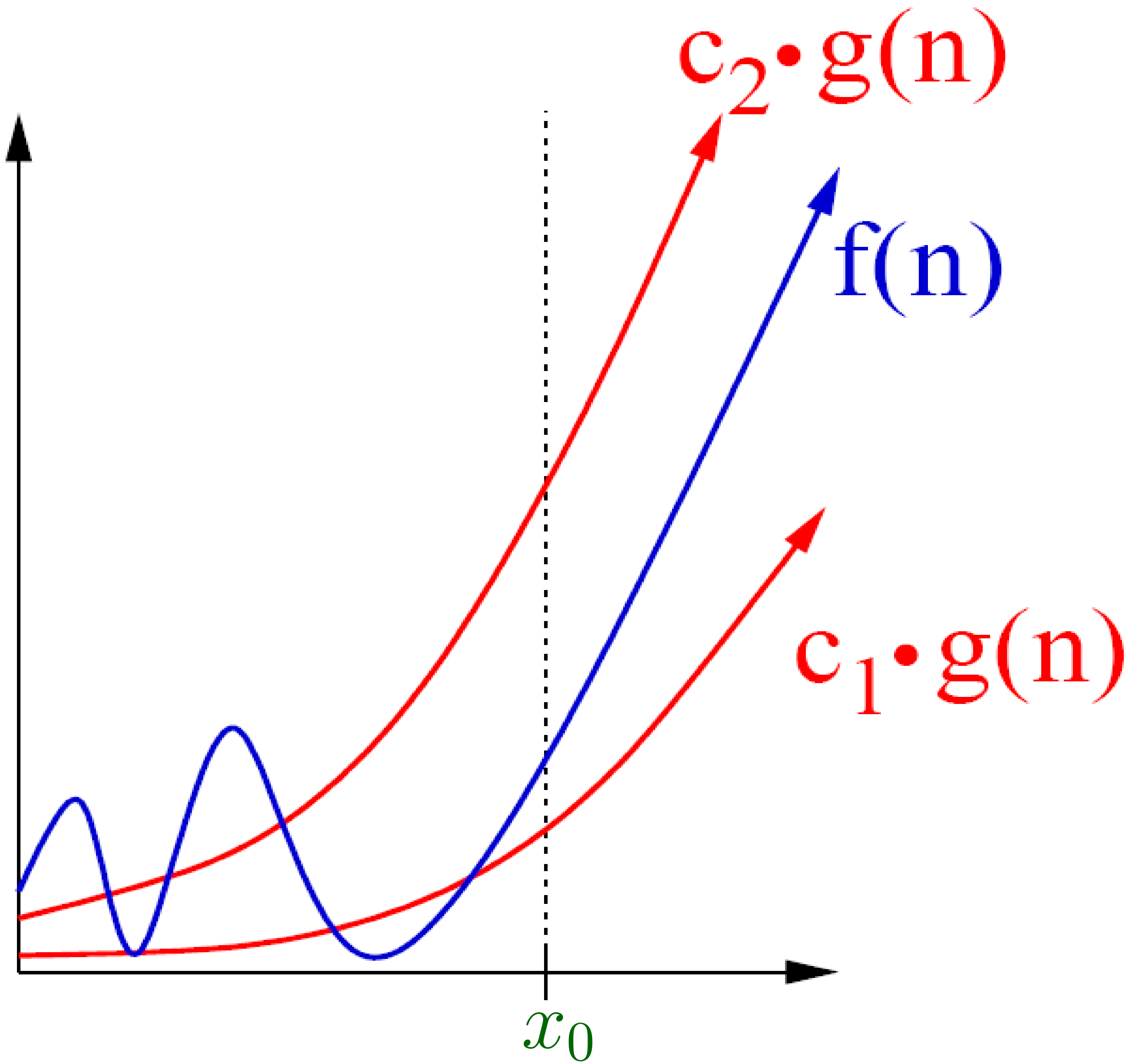
Two functions $f(n), g(n)$ have the same order of growth if
$f(n) = O(g(n))$ and $g(n) = O(f(n))$.

In this case we say

$$f(n) = \Theta(g(n))$$

which is the same as

$$g(n) = \Theta(f(n))$$


c2•g(n)
f(n)
c1•g(n)
x0

Examples ($f(n) = \Theta(g(n))$):

- $3n^2 + 4n = \Theta(n)$? No
- $3n^2 + 4n = \Theta(n^2)$? Yes
- $3n^2 + 4n = \Theta(n^3)$? No, but $O(n^3)$
- $n/5 + 10n\log n = \Theta(n^2)$? No, but $O(n^2)$
- $n^2/5 + 10n\log n = \Theta(n\log n)$? No
- $n^2/5 + 10n\log n = \Theta(n^2)$? Yes