BIG-O NOTATION AKA THETA NOTATION

M	1	2	3	···	···	n-1	n
1	5	6	72			101	-7
2	-2	13	-8	- - -		56	0
3	1						
⋮		⋮		⋱		⋮	
⋮							
n-1	29			- - -			
n	42						

(assume n is a multiple of 100)

for each row i $(1 \leq i \leq n)$

1) print $M[i,1]$ 5 times

2) for each column j $(1 \leq j \leq n)$
   if j is even, print $M[i,j]$

3) if i is a multiple of 100, print all of M

How many times do we print something?

$$5n + \frac{1}{2}n^2 + \frac{n}{100}\cdot n^2 = O(n^3)$$

$$5n + \frac{1}{2}n^2 + \frac{1}{100}\cdot n^3 \quad = O(n^3)$$

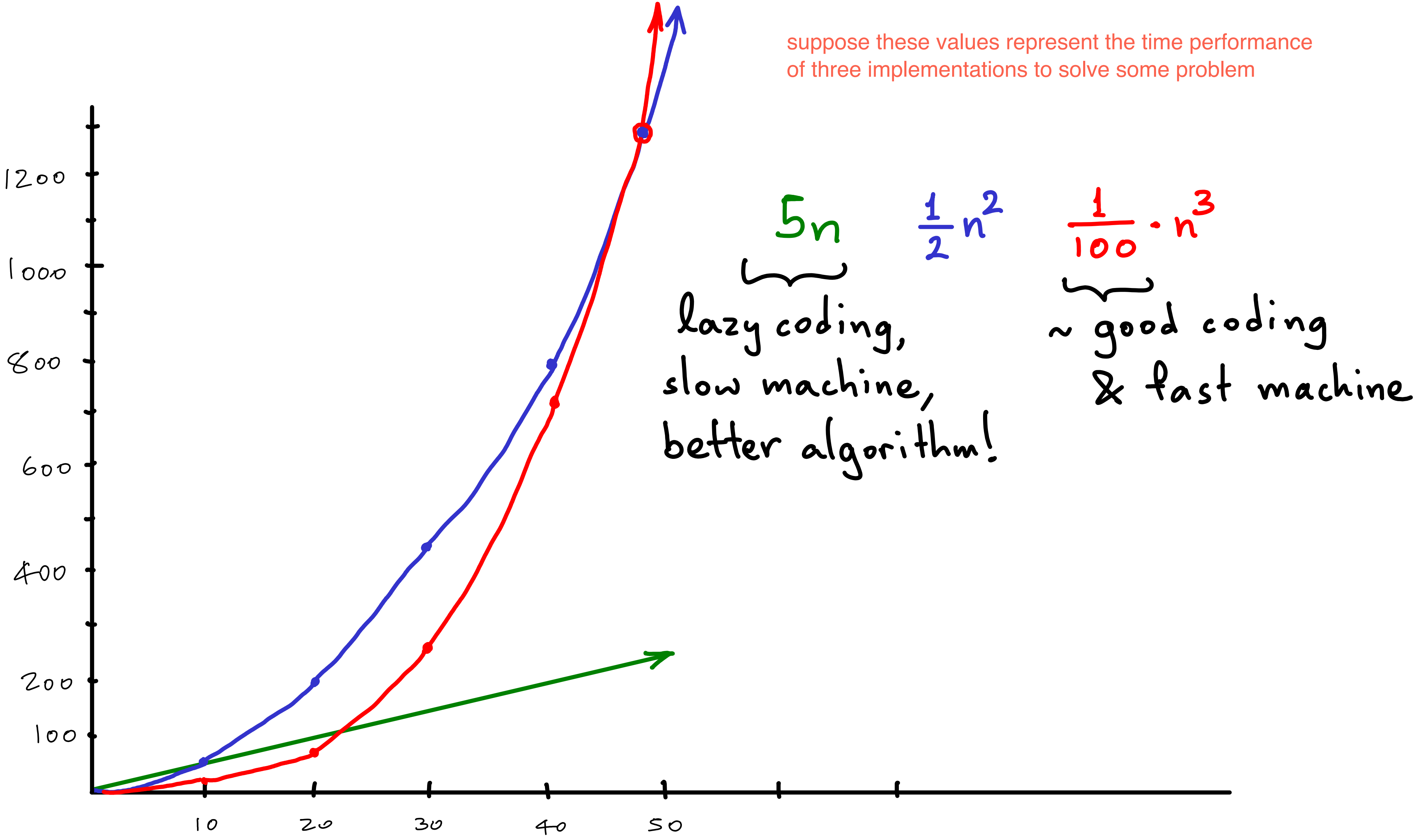
We care only about the dominant term, as $n \to \infty$

↳ in fact we only care about the "pure" component involving n.

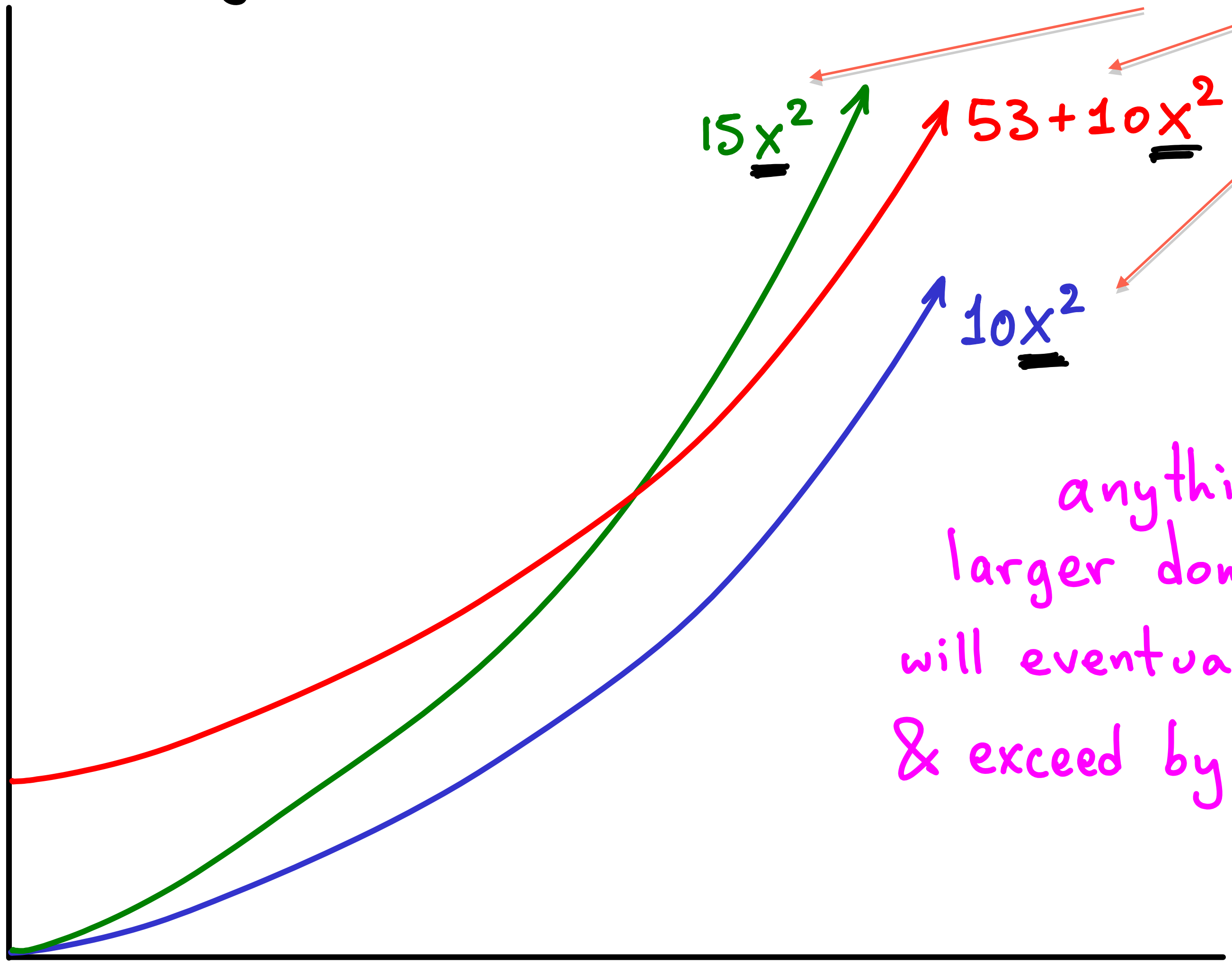
Why ?

- none of the lower terms or leading constants matter compared to the part we keep, as n grows large.
- often we don't even know what the leading constants are or they depend on a particular language/platform/OS/etc but the general behavior in terms of n doesn't change.

suppose these values represent the time performance of three implementations to solve some problem

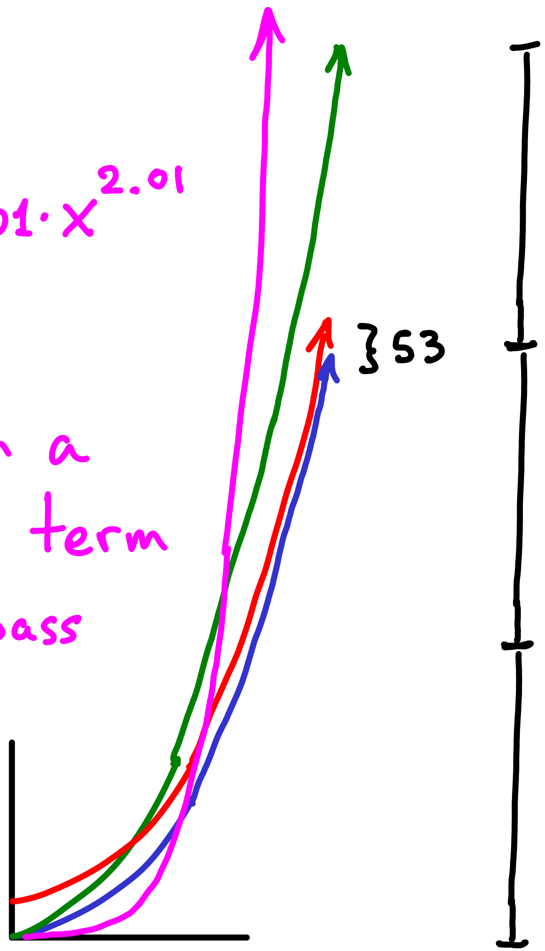


For any x, these stay within a constant additive or multiple factor



$0.00001 \cdot x^{2.01}$

anything with a larger dominating term will eventually surpass & exceed by far



Formally, $f(n) = O(g(n))$ —OR— $f(n) \in O(g(n))$

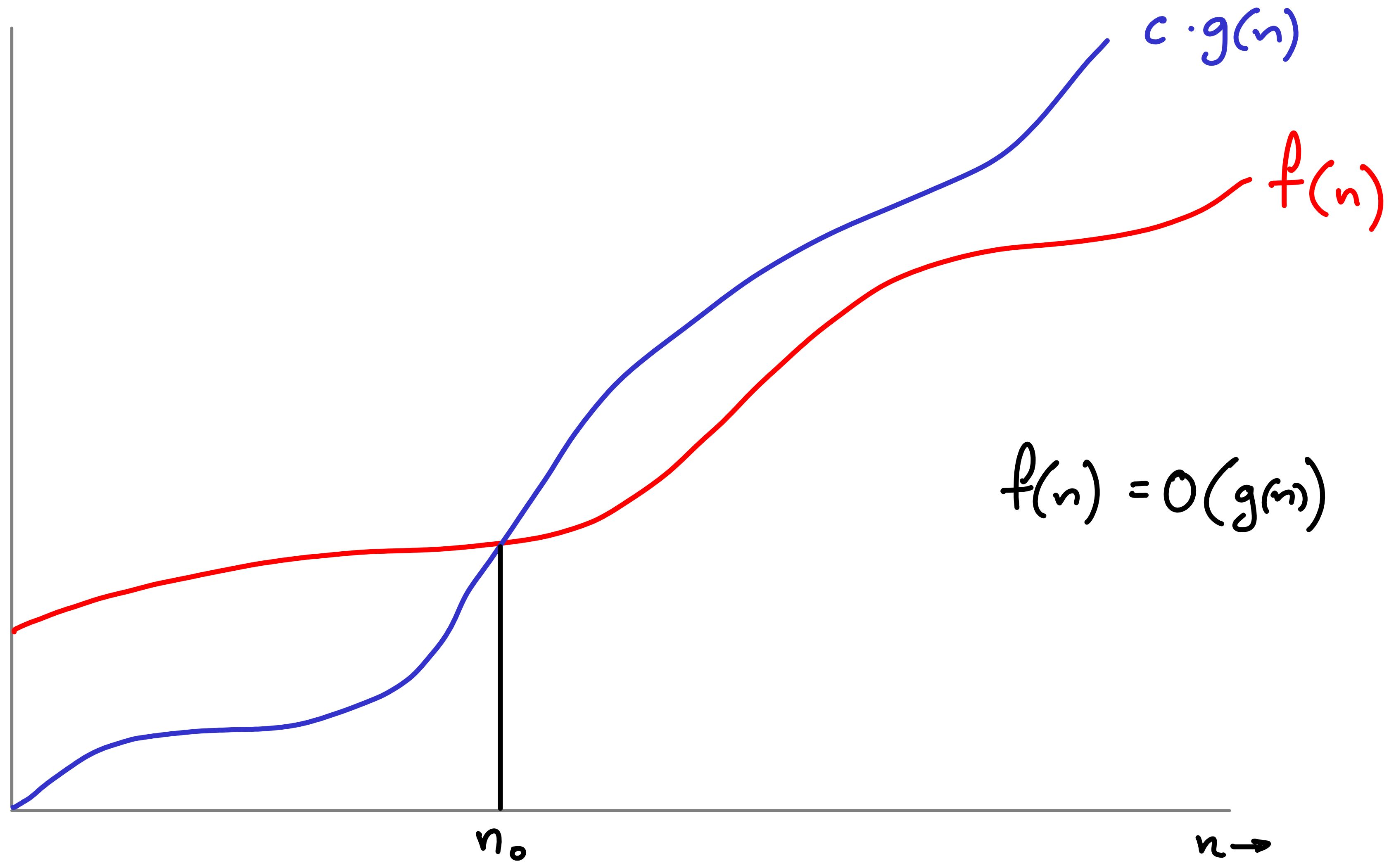
if for all $n > n_0$
there is a constant c } such that $0 \leq f(n) \leq c \cdot g(n)$

upper bound.

(n_0 is also a constant)

$g(n)$ → is a simplification of $f(n)$.
→ is an asymptotic upper bound for $f(n)$.

within a constant factor,
for large n


$c \cdot g(n)$
$f(n)$
$f(n) = O(g(n))$
n_0
$n \rightarrow$

Polynomials : $a + bn + cn^2 + dn^3 \dots + zn^k = O(n^k)$

It is important that a, b, c, k, etc are constants

$a \cdot n^x + b \cdot n^y = O(n^x)$ [because $bn^y = O(n^x)$]

Assume $x \geq y$. a & b are constants, so don't matter.

Logarithms : $50 \cdot \log n + 10 \log^5 n + n^{0.1} = O(n^{0.1})$

"weaker" than polynomial [note: $50 \log n = O(\log^5 n)$]

Exponential : $100 \cdot n^{50} + 3^n + 40 \cdot 2^n = O(3^n)$

"stronger" than polynomial [note: $40n^2 = O(3^n)$]

$$\left(50 \cdot \log n + 10 \log^5 n + n^{0.1}\right) \cdot \left(100 \cdot n^{50} + 3^n + 40 \cdot 2^n\right) =$$

$$O(n^{0.1}) \cdot O(3^n) = O(n^{0.1} \cdot 3^n)$$

- is $5n^2 = O(n^3)$? yes. but a better answer is $O(n^2)$
- is $n^3 = O(5n^2)$? No. & we shouldn't even use $O(5n^2)$

(don't put unnecessary constants in big-O)

$$\log(n!) = O(\log(n^n)) = O(n\log n)$$

$$\underset{\text{const}}{a} = O(1) = O(\log^{b} n) = O(n^{c}) = O(n^{c}\log^{d} n) = O(k^{n}) = O(n!) = O(n^{n})$$

$b>0 \qquad c>0 \qquad d>0 \qquad k>1$

$$O(1) \downarrow O(n^{0})$$

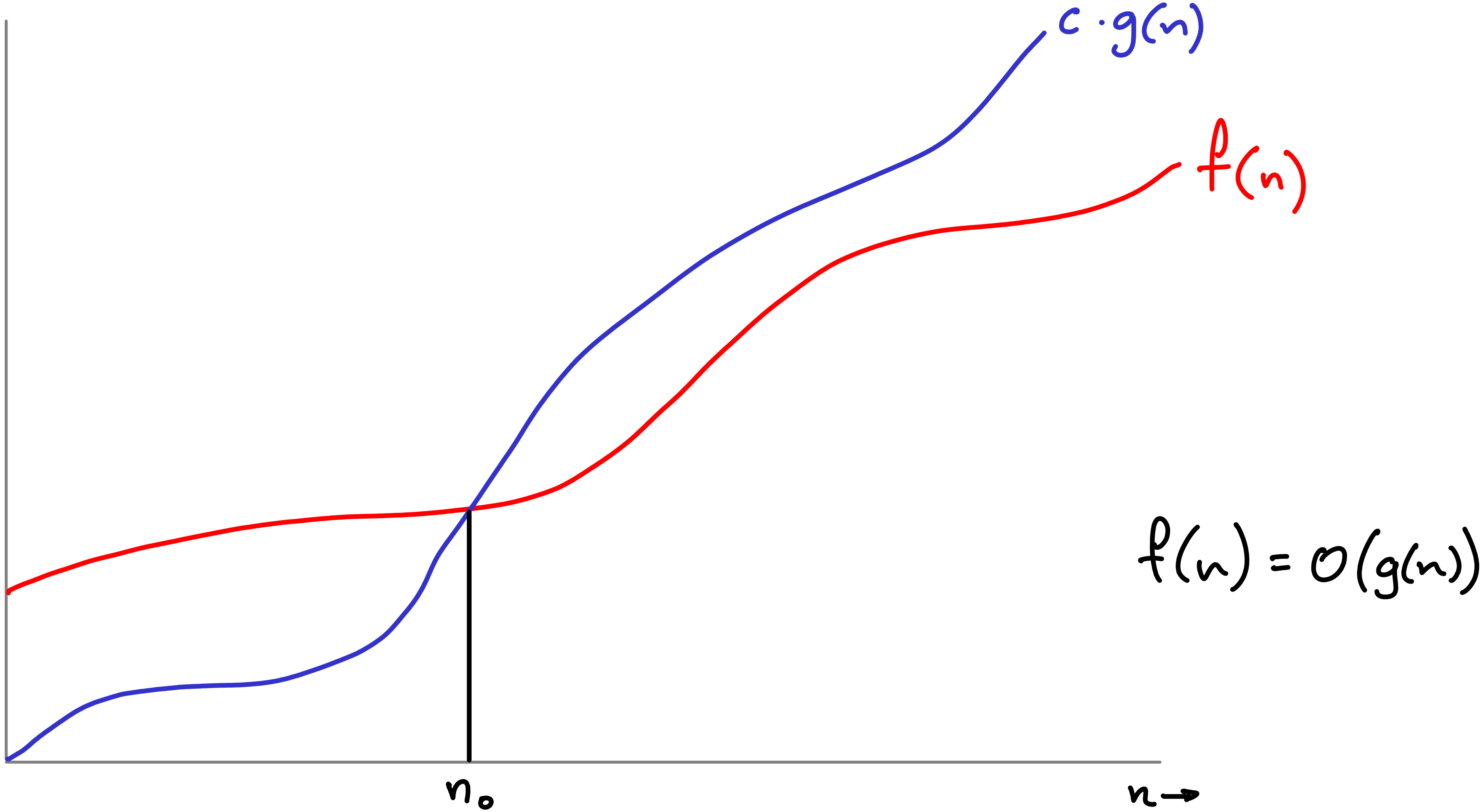
RECAP: $f(n) = O(g(n))$ means $f(n) \le \underbrace{c \cdot g(n)}_{\text{upper bound}}$ (for all $n > n_0$)

We can also get asymptotic lower bounds this way

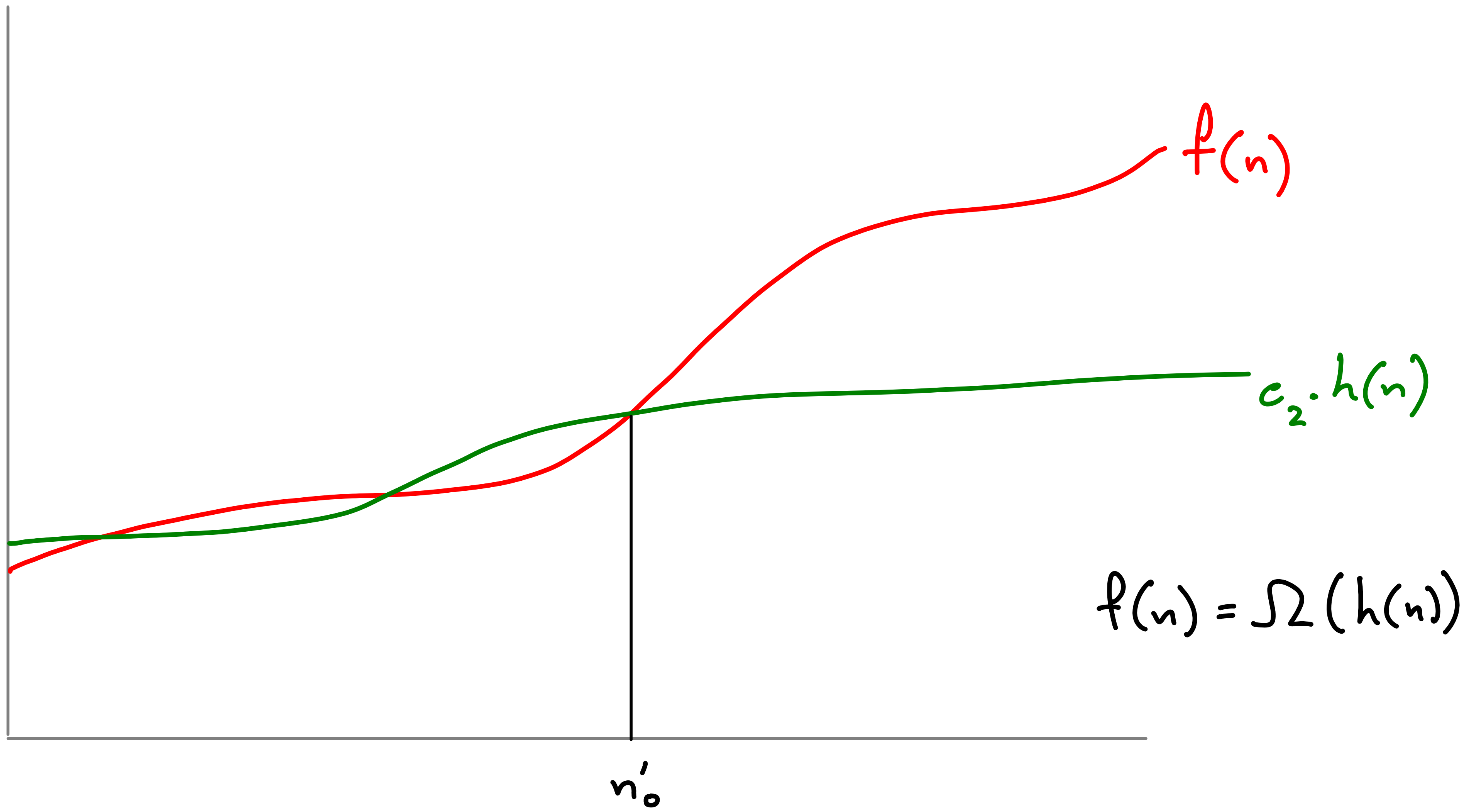
Suppose $f(n) \geq c \cdot g(n)$ for some constant c,
& for all $n > n_0$

Then $f(n) = \Omega(g(n))$

e.g. $\frac{1}{5}n^2 = \Omega(n)$ [but also $\Omega(n^2)$]

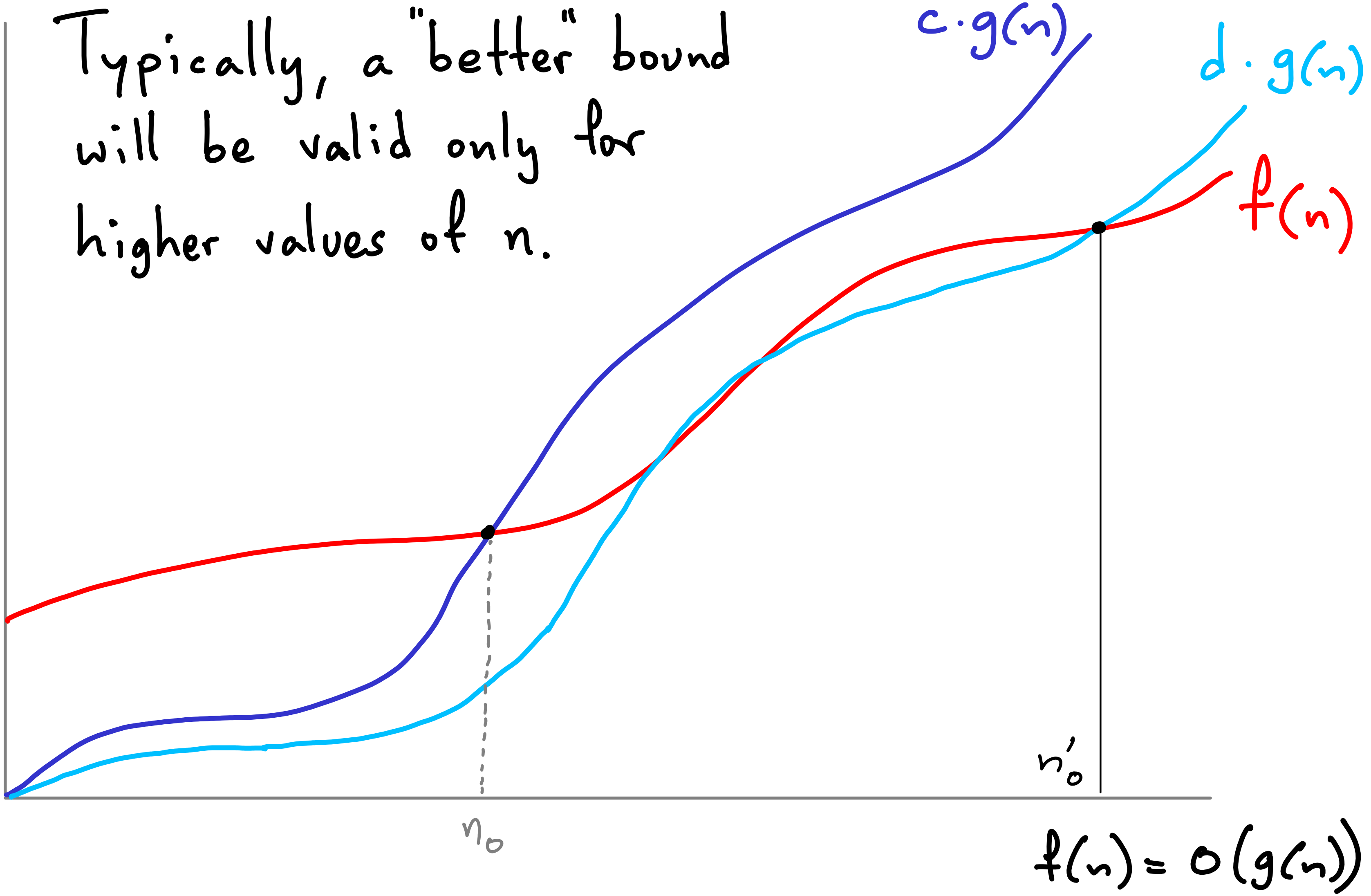


$c \cdot g(n)$
$f(n)$
$f(n) = O(g(n))$
n_0
$n \rightarrow$



$f(n)$
$c_2 \cdot h(n)$
n_0'
$f(n) = \Omega(h(n))$

Typically, a "better" bound will be valid only for higher values of n.



$d < c$

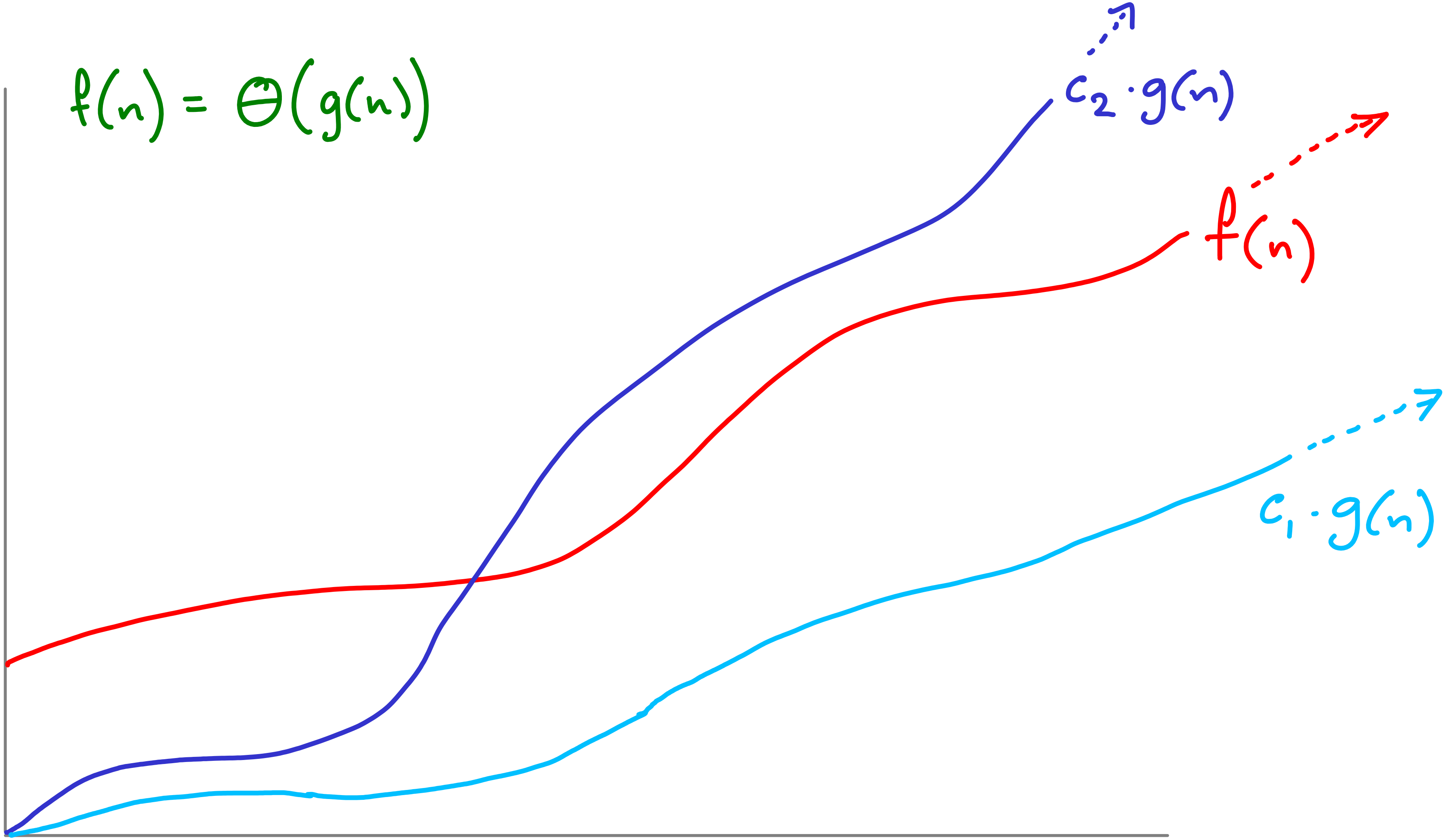
$\downarrow$

better

if $f(n) = O(g(n))$

and $f(n) = \Omega(g(n))$

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



$f(n) = \Theta(g(n))$
$c_2 \cdot g(n)$
$f(n)$
$c_1 \cdot g(n)$

Prove $\frac{1}{2}n^2 - 3n = \Theta(n^2)$

↳ prove $= O(n^2)$ → find c_1 & n_1 s.t. $\frac{1}{2}n^2 - 3n \leq c_1 \cdot n^2$ for $n > n_1$

$\frac{1}{2}n^2 - 3n < \frac{1}{2}n^2 \Rightarrow c_1 = \frac{1}{2}$ and $n_1 = 1$ work

[or more] [or more]

↳ prove $= \Omega(n^2)$ → find c_2 & n_2 s.t. $\frac{1}{2}n^2 - 3n \geq c_2 \cdot n^2$ for $n > n_2$

$\frac{1}{2} - \frac{3}{n} \geq c_2$ → for $n \leq 6$, $c_2 \leq 0$: invalid

$\Rightarrow$ choose $n_2 = 7$ & $c_2 = \frac{1}{14}$

[or less]

Combine: for $n \geq 7$ we have c_1 & c_2 s.t.

$$c_2 \cdot n^2 \leq \frac{1}{2}n^2 - 3n \leq c_1 \cdot n^2$$

Note: if we choose $n_2 = 30$ then $c_2 \leq 0.4$ works.
In fact we can get $c_2 \to \frac{1}{2}$ if $n_2 \to \infty$

Prove $6n^3 \neq \Theta(n^2)$

if it were true, then $\underbrace{c_1 n^2 \leq 6n^3}_{\Omega} \leq c_2 n^2$ for $n \geq n_0$, with $\overbrace{6n^3 \leq c_2 n^2}^{O}$

$6n^3 \geq c_1 \cdot n \rightarrow$ trivially true for $n \geq 1$ & $c_1 = 6$

is $6n^3 = O(n^2)$? $\rightarrow$ $6n^3 \leq c_2 n^2$?

$\boxed{\text{NO}}$

$6n \leq c_2$?

$n \leq \frac{c_2}{6}$ } NO.

Whatever constant c_2 we choose, n will eventually surpass it.

$\log(n!) = O(n \log n)$ $\qquad\qquad\qquad\qquad$ c $\qquad\qquad$ n_0

$\log(n!) \leq \log(n^n) = n \log n \quad \Rightarrow \quad \log(n!) \leq 1 \cdot n \log n$ for $n \geq 1$

$\log(n!) = \Omega(n \log n)$

$$\log(n!) = \log(n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdots \cdots 3 \cdot 2 \cdot 1)$$

$$= \log(\underbrace{n \cdot 1} \cdot \underbrace{(n-1) \cdot 2} \cdot \underbrace{(n-2) \cdot 3} \cdot \underbrace{(n-3) \cdot 4} \cdots \cdots \sim \underbrace{(n-\tfrac{n}{2}) \cdot (n-\tfrac{n}{2})})$$

$$\geq \log(n \cdot n \cdot n \cdot n \cdot \cdots n)$$

$$= \log(n^{n/2}) \quad \text{(assume } n\text{: even} \quad \text{otherwise } \lfloor \tfrac{n}{2} \rfloor \text{)} \quad \Rightarrow \quad \log(n!) \geq \frac{n}{2} \log n$$

---

so $\frac{1}{2} n \log n \leq \log(n!) \leq n \log n$

---

in fact, Stirling's approximation: $\ln(n!) = n \cdot \ln(n) - n + O(\ln(n))$