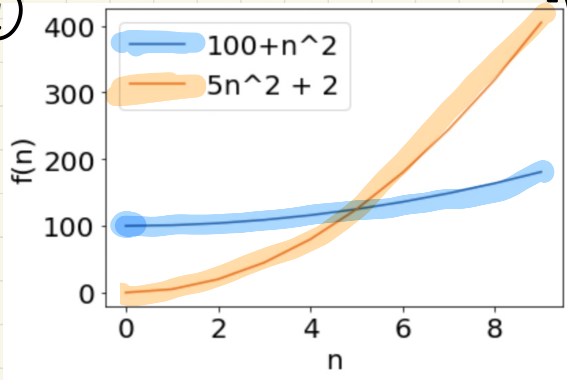
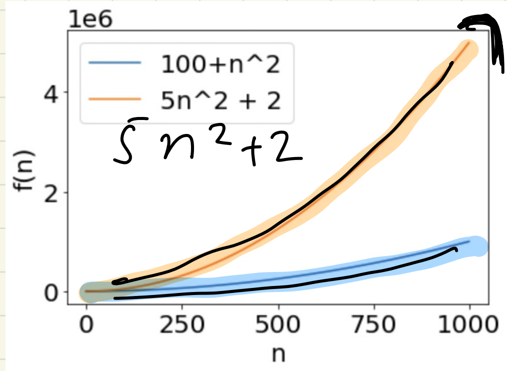
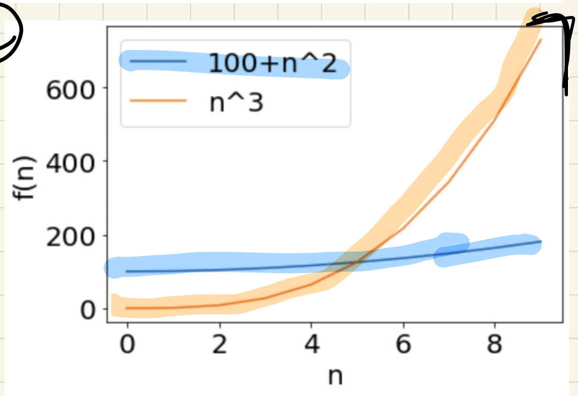
which function is "smaller (or equal)"?

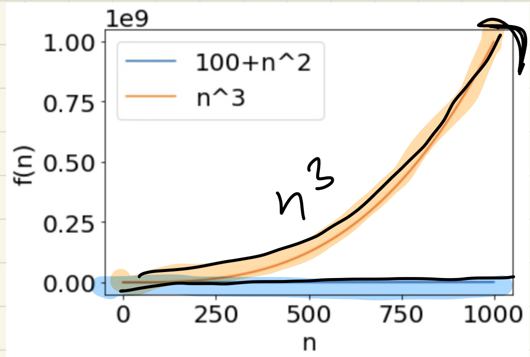
1



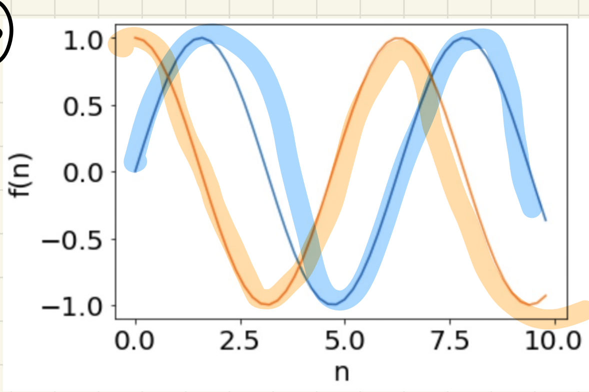


2





3



In CS, we focus on "grows no faster than" as an approximation to "smaller than or equal to."

<u>Def</u> $f: \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$, $g: \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$

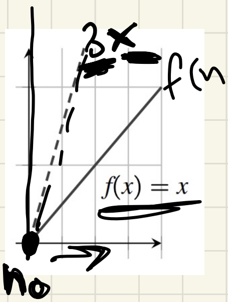
we say that $f = O(g)$ "f is big O of g" if $\exists\, c>0,\ n_0 \geq 0$ s.t. $\forall n \geq n_0 : f(n) \leq c \cdot g(n)$.

Note: $f = O(g)$ is standard, but it uses "=" to mean "has the property"

To prove $f(n) = O(g(n))$, we need to construct n_0, c s.t. $\forall n \geq n_0 : f(n) \leq c \cdot g(n)$.

To prove $f(n) \neq O(g(n))$, we need to show that $\forall n_0, c, \exists n \geq n_0 : f(n) > c \cdot g(n)$.

Examples: all of the following functions f are $O(n)$. (all grow no faster than n)

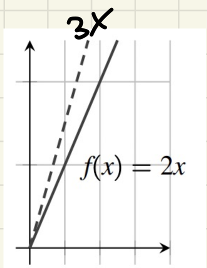


WTS $f(n) = n$ is $O(n)$.

let $n_0 = 0$, $c = 3$.

$\forall n \geq 0 : n \leq 3n$

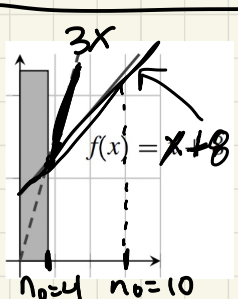
$f(n) \leq g(n) \cdot c$ $\square$



WTS $f(n) = 2n$ is $O(n)$.

let $n_0 = 5$, $c = 1$

$\underline{\forall n \geq 5} : \underline{f(n) = 2n} \leq 1 \cdot n$ $\square$



WTS $f(n) = n+8$ is $O(n)$.

does $n_0 = 0$, $c = 3$ work? no

would need $\forall n \geq 0 : n+8 \leq \underline{3n}$. doesn't work $n=0$

Say I want to choose $c=3$. what n_0 can I use?

smallest n_0? plug in: $n_0+8 = 3n_0$

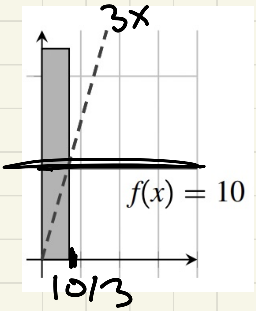
$8 = 2n_0$

$4 = n_0$

any $n_0 \geq 4$ would work.

consider $n_0 = 4$, $c = 3$.

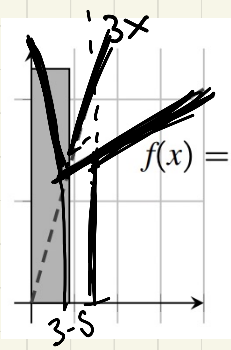
$\forall n \geq n_0 = 4$: $n+8 \leq 3n$. $\square$



WTS $f(n) = 10$ is $O(n)$ $\quad 10 = 3n_0$

let $n_0 = 3.4$, $c = 3$

$\forall n \geq 3.4 : 10 \leq 3n$ $\square$



$$f(x) = \begin{cases} 25 - x^2 & \text{if } x < 3.5 \\ 0.5x + 11 & \text{if } x > 3.5 \end{cases}$$

WTS $f(n) = O(n)$.

let $n_0 = 5$, $c = 3$.

check: $0.5(5) + 11 = 13.5$

$3 \cdot g(5) = 3 \cdot 5 = 15$

$\forall n \geq 5 : f(n) \leq 3 \cdot n$ $\square$

Example: $n^3 \neq O(n^2)$.

Proof: WTS $\forall c > 0,\ n_0 \geq 0 : \exists n \geq n_0 : n^3 > c \cdot n^2$.

We prove this by showing how to construct n for any c, n_0.

Let $c > 0$, $n_0 \geq 0$. We need $n \geq n_0$ s.t. $n^3 > c \cdot n^2$. Let $n = (c+1)$. Then $n^3 = (c+1)^3$ and $c \cdot n^2 = c((c+1)^2$. Notice that $(c+1)^3 > c \cdot (c+1)^2$ because $c > 0$, so $n^3 > c \cdot n^2$.

We have n s.t. $n^3 > cn^2$, but we also need that $n \geq n_0$. Let $n = \max\{n_0, c+1\}$.

$\square$

Logarithms for positive real number $b \neq 1$ and real number $x > 0$, $\log_b x$ is the real number y s.t. $b^y = x$.