$f(n)$ is $O(g(n))$ if, for sufficiently large n, $f(n)$ is upper-bounded by a constant multiple of $g(n)$.

there exist

$f(n)$ is $O(g(n))$ if $\exists\ n_0 \geq 0,\ c > 0$ such that $\forall\ n \geq n_0 : f(n) \leq c \cdot g(n)$.

for all

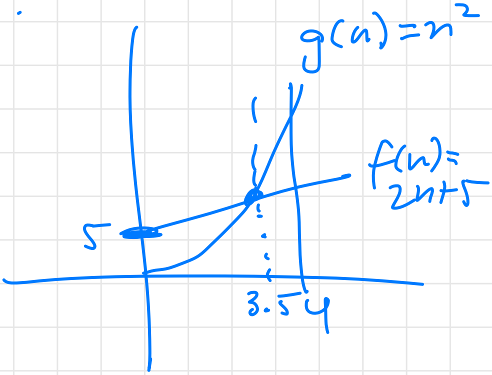
Claim $\underbrace{2n+5}_{f(n)}$ is $O(\underbrace{n^2}_{g(n)})$.

Proof: To show that $2n+5$ is $O(n^2)$, we must show that $\exists\ n_0 \geq 0,\ c > 0$ such that:

$\forall n \geq n_0 : \quad 2n+5 \leq cn^2.$

Consider $c=1$, $n_0 = 4$. It is true that

$2n+5 \leq n^2$ for all $n \geq 4$.



So $2n+5$ is $O(n^2)$.

Claim n^2 is $O(2n+5)$.

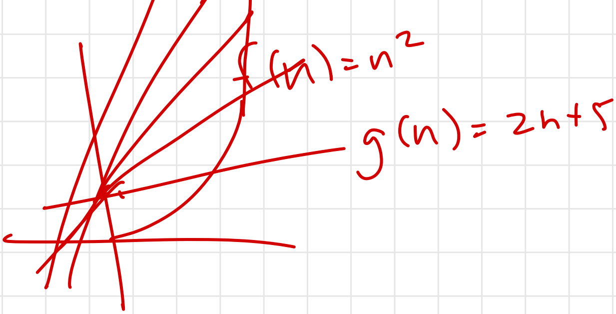
Proof: We need $n_0 \geq 0$, $c > 0$ such that

$\forall n \geq n_0 : \quad n^2 \leq c\,(2n+5).$

<u>claim</u>: n^2 is <u>not</u> $O(2n+5)$.

<u>Proof</u>: We need to show that there do <u>not</u> exist $n_0 \geq 0$, $c > 0$ such that

$$\forall n \geq n_0 : \quad n^2 \leq c \cdot (2n+5).$$



$n^2 + \underline{\log n}$ is $O(n^2)$

Properties of big O:

- constant don't matter
- drop lower order terms
- bases of logs don't matter
- bases of exponents do matter $\quad 2^n \quad 3^n$

Claim If $f(n) = O(g(n) + h(n))$, then $f(n) = O(\max(g(n), h(n)))$.

ex $n^2 + n = O(n^2 + n)$ $f(n) = O(f(n))$

$n^2 + n = O(\max(n^2, n))$

$= O(n^2)$

Proof Assume $f(n) = O(g(n) + h(n))$.

This means that there is a n_0, c so that

$$\forall n \geq n_0: \quad f(n) \leq c \cdot [g(n) + h(n)]$$

$$\leq c \cdot [\max(g(n), h(n)) + \max(g(n), h(n))]$$

$$\leq c \cdot [2 \max(g(n), h(n))]$$

$$\leq 2c \cdot \max(g(n), h(n))$$

choose $n_0' = n_0$ $c' = 2c$

So $f(n) = O(\max(g(n), h(n)))$