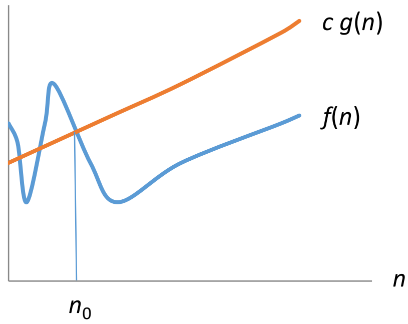
To us, the function f will usually be the running time of an algorithm that we analyze

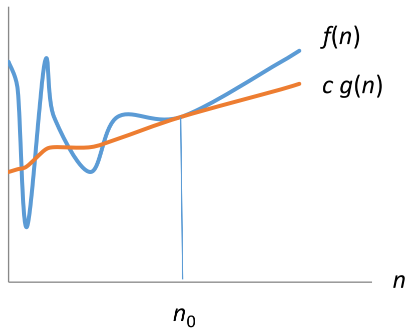
"O-notation" (Asymptotic notation)

O ("big Oh") – upper limit



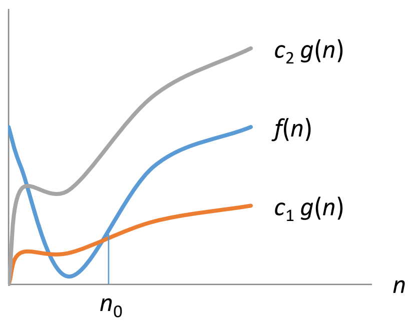
Notation	$\boldsymbol{f(n) = O(g(n))}$
Intuition	$\boldsymbol{f}$ **is smaller than** $\boldsymbol{g}$
When $n \to \infty$	$\boldsymbol{f(n) \leq c \cdot g(n)}$
Definition	$\boldsymbol{\exists (c > 0), n_0 : \forall (n > n_0)\ f(n) \leq c \cdot g(n)}$

Ω ("Omega") – lower limit



Notation	$\boldsymbol{f(n) = \Omega(g(n))}$
Intuition	$\boldsymbol{f}$ **is larger than** $\boldsymbol{g}$
When $n \to \infty$	$\boldsymbol{f(n) \geq c \cdot g(n)}$
Definition	$\boldsymbol{\exists (c > 0), n_0 : \forall (n > n_0)\ f(n) \geq c \cdot g(n)}$

Θ ("Theta") – "as"



Notation	$\boldsymbol{f(n) = \Theta(g(n))}$
Intuition	$\boldsymbol{f}$ **grows like** $\boldsymbol{g}$**, i.e. between** $\boldsymbol{c_1 \cdot g}$ **and** $\boldsymbol{c_2 \cdot g}$
When $n \to \infty$	$\boldsymbol{c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)}$
Definition	$\boldsymbol{\exists (c_1, c_2 > 0), n_0 : \forall (n > n_0)\ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)}$

o ("little oh")[1]

Notation	$\boldsymbol{f(n) = o(g(n))}$
Intuition	$\boldsymbol{f}$ **is a lot smaller than** $\boldsymbol{g}$
When $n \to \infty$	$\boldsymbol{f(n) \ll g(n)}$
Definition	$\boldsymbol{\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0}$

---

[1] *O*-notation might not be *asymptotically tight*: $2n^2 = O(n^2)$ is asymptotically tight, but $2n = O(n^2)$ is not. We use *o*-notation (little oh) to indicate that our analysis is "un-tight".