Describing:
- what
- how
- why
- fast
Comparing running times is not easy...

Everything fast if n=2

so focus on large n...

& ignore constant factors

→ asymptotic notation
Big-oh:

function \( f(n) : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0} \)

\( f \) time on input size \( n \)

\( g(n) : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0} \)

Define the set

\( O(g(n)) = \{ f(n) : \text{there exist pos. constants } c, n_0 \text{ s.t. } 0 \leq f(n) \leq c g(n) \text{ for all } n \geq n_0 \} \)
\( O(g(n)) \) is all functions that are \( \leq \) a constant multiple of \( g(n) \) when \( n \) is "large".

\[ c \cdot g(n) \geq n_0 \]
big-Oh is like $\leq$

$256 \cdot n \in O(n) \quad (n_0 = 0 + c = 256)$

$n \in O(n^2) \quad (n_0 = 256 + c = 1)$

claim a running time in $O(n^2 \log n)$ when it is in $O(n^2)$, still "correct"
Lower bound: $6 + g^\Omega$

$$\Omega(g(n)) = \{ f(n): \text{there exist pos. constants } c, n_0, \text{ s.t. } 0 \leq g(n) \leq f(n) \text{ for all } n \geq n_0 \}$$

Exact: $\Theta$

$$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$$

$f(n)$ is "asymptotically tight" to $g(n)$
$o(g(n))$: informally:

functions in $O(g(n))$

but not in $\Theta(g(n))$

Merge sort runs in $O(n \log n)$. Is there one in $o(n \log n)$?

"No"

$\omega(g(n))$: tight lower bound
Algebra

Transitivity: If $f(n) \in O(g(n)) + g(n) \in O(h(n))$, then $f(n) \in O(h(n))$

true for $O, \Omega, \Theta, \omega$

Reflexivity: $f(n) \in O(f(n))$

Symmetries: $f(n) \in \Theta(g(n))$ iff $g(n) \in \Theta(f(n))$

$f(n) \in O(g(n))$ iff $g(n) \in \Omega(f(n))$
\[ f(n) \in o(g(n)) \iff \text{iff} \]
\[ g(n) + o(f(n)) \uparrow \text{iff} \]
\[ \text{iff (\iff)} \]

"Algebra": Suppose \[ f_1(n) \in \Theta(g_1(n)) \]
\[ + f_2(n) \in \Theta(g_2(n)) \]
\[ c \cdot f_1(n) \in \Theta(f_1(n)) \text{ for constant} \]
\[ f_1(n) + f_2(n) \in \Theta(g_1(n) + g_2(n)) \]
\[ f_1(n) + f_2(n) \in \Theta(\max \{ g_1(n), g_2(n) \}) \]
\[ f_1(n) \cdot f_2(n) = \Theta(g_1(n) \cdot g_2(n)) \]
If you have asymptotic notation in an algebraic expression, ...

then for any choices of functions for each "written" set on the left hand side (LHS) there should exist choices on RHS to make the expression true.
\[ f(n) = O(g(n)) \] is true if and only if \( f(n) \in \Theta(g(n)) \).

\[ 5n^2 + 1000n = 5n^2 + O(n) \]

Then someone writes

\[ 5n^2 + O(n) \leq \text{[something]} \]
What is the running time if each line takes \( O(1) \) time?

If \( m = n \)?

\( k \leq 2n \), so inner loop runs at most \( 2n+1 = O(n) \) times.

Inner loop takes \( O(n \cdot O(1)) = O(n) \).

Whole outer loop iteration
takes \( O(n) + O(1) + o(1) \)  
= \( O(n) \) time

whole alg takes  
\( O(1) + 2n \cdot O(n) + o(1) = O(n^2) \) time

i.e., its a doubly nested for loop over \( O(n) \) values
Favorite Functions

$f(n)$ is polynomially bounded if $f(n) = O(n^k)$ for some constant $k$.

Main first goal for times

$n^{k_1} = o(n^{k_2})$ when $k_1 < k_2$.

So aim for small exponents.
exponential: \( a^n \)

\( n^k = O(a^n) \) for any constant \( k \) and constant \( a > 1 \).

\( a^n = o(c^n) \) for constants \( c > a > 1 \).
polylogarithmically bounded

\[(\log_b n)^{\ell} = o(n^k)\] for constants \(b > 1, \ell, k > 0\).

Example: \(n^2 = \omega\left(n^{1.9} (\log_{\sqrt{5}} n)^{1000\cdots} \right)\)
\[ \log n := \log_{10} n \]
\[ \lg n := \log_2 n \]
\[ \ln n := \log_\mathrm{e} n \]

\[
\hat{\approx} \text{Euler's number} \approx 2.72
\]

\[ \log_b \ln : (\log_b n)^l \]

\[ a \log_b n = n \uparrow \text{write like this} \]
\[
\log_b n = \frac{\log n}{\log b}
\]

so, if \( b \) is constant,
\[
\log_b n = \Theta(\log n)
\]

\[
n \log_3 5 = o(n^{\log_5 3})
\]

\[
\text{oh no!}
\]