Asymptotic Analysis.

"How quickly do things grow relative to other things?"

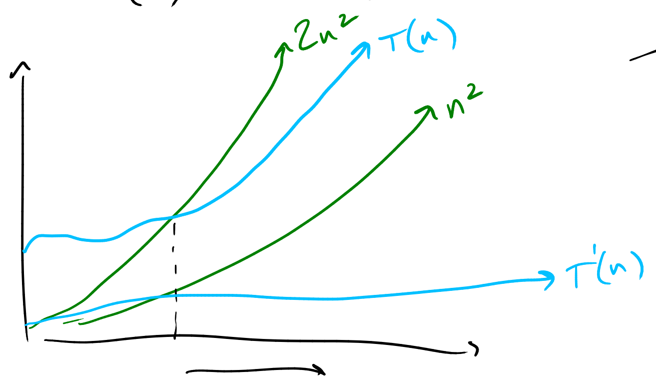
Why use it?

- Often quicker than actually running + timing.
- Analyze all possible input sizes at once.
- Insulate ourselves from noise due to hardware, software, etc.

- big-O is kind of like $\leq$
  big-Ω is " " " $\geq$
  big-Θ " " " $=$

any constant.

e.g. "$T(n)$ is $O(n^2)$" means "$T(n)$ is $\leq c \cdot n^2$, eventually".

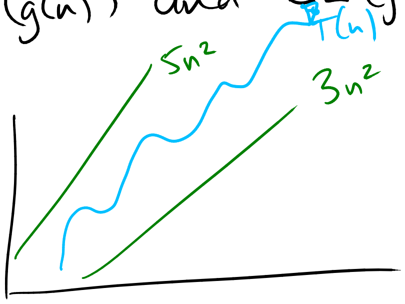


Def'n $T(n)$ is $O(g(n))$ iff there exists some natural number n_0 and real number $c > 0$ such that for all $n \geq n_0$,

$$T(n) \leq c \cdot g(n).$$

$\geq$ is def'n of $\Omega(g(n))$.

Def'n If $T(n)$ is both $O(g(n))$ and $\Omega(g(n))$, we say $T(n)$ is $\Theta(g(n))$.



$3n^2 + 5n - 7$ is $\Theta(n^2)$, but showing this according to defn would be very tedious. But we can use limits!

Idea is to look at $\lim_{n \to \infty} \frac{T(n)}{g(n)}$.

Thm.
- If $0 \leq \lim_{n\to\infty} \frac{T(n)}{g(n)} < \infty$, then $T(n)$ is $O(g(n))$.
- If $0 < \lim_{n\to\infty} \frac{T(n)}{g(n)} \leq \infty$, then $T(n)$ is $\Omega(g(n))$.
- If $0 < \lim_{n\to\infty} \frac{T(n)}{g(n)} < \infty$, then $T(n)$ is $\Theta(g(n))$.

eg. $$\lim_{n\to\infty} \frac{3n^2 + 5n - 7}{n^2} = \lim_{n\to\infty} \frac{3\cancel{n^2}}{\cancel{n^2}} + \lim_{n\to\infty} \frac{5\cancel{n}}{n^{\cancel{2}}} + \lim_{n\to\infty} \frac{-7}{n^2}$$
$$= 3 + 0 + 0 = 3.$$

Properties of $O / \Omega / \Theta$.

- $k \cdot O(f(n)) = O(f(n))$ [If k is a constant — ie. does not depend on n].
  (ie. running algorithm k times, or on slower or faster computer).

- $O(f(n)) + O(g(n)) = O(\max(f(n), g(n)))$
  eg. $O(n) + O(n^2) = O(n^2)$.
  (running algorithms in sequence).

- $O(f(n)) \times O(g(n)) = O(f(n) \times g(n))$
  (running nested algorithms).

eg. $O(3n^2 + 5n - 7) = O(3n^2) + O(5n) - O(7)$
$= 3 \cdot O(n^2) = O(n^2)$.