Time Complexity

Sorting $100n^2$ operations

big-O $f(n)$, $g(n)$

upper bound	$g(n) = O(f(n))$	$\lim_{n\to\infty} \frac{g(n)}{f(n)}$	$\leq c$ (for some constant)
lower ~	$g(n) = \Omega(f(n))$		$\geq c$
same	$g(n) = \Theta(f(n))$		$= c$
loose upper	$g(n) = o(f(n))$	$\lim_{n\to\infty} \frac{g(n)}{f(n)}$	$= 0$
loose lower	$g(n) = w(f(n))$		$= \infty$

$100n^2 = O(n^2)$

$2n^3 + 100n^2 = O(n^3)$

$n\log n = o(n^2)$

$\frac{n^3}{\log\log n} = o(n^3)$

$2^n = o(n^n)$

$n^{\log_2 n} = 2^{\log_2^2 n}$

$n! \sim \Theta\left(\left(\frac{n}{e}\right)^n\right) = 2^{n\log_2 n}$

diff notations:

$n^2 \in O(n^3)$

$n^2 \leq O(n^3)$

$10^{10^{10}} n^2 = O(n^2)$ $\quad n^3 = O(n^3)$

in theory, $10^{10^{10}} n^2$ is better

but in practice, n^3 is faster

good algorithms

$poly(n) = O(n^c) \rightarrow$ good

$2^n \rightarrow$ not good