Big-O

runtime of a program: how long it takes to run on an input of size n.

e.g.) a program's runtime is n^2, n size input.

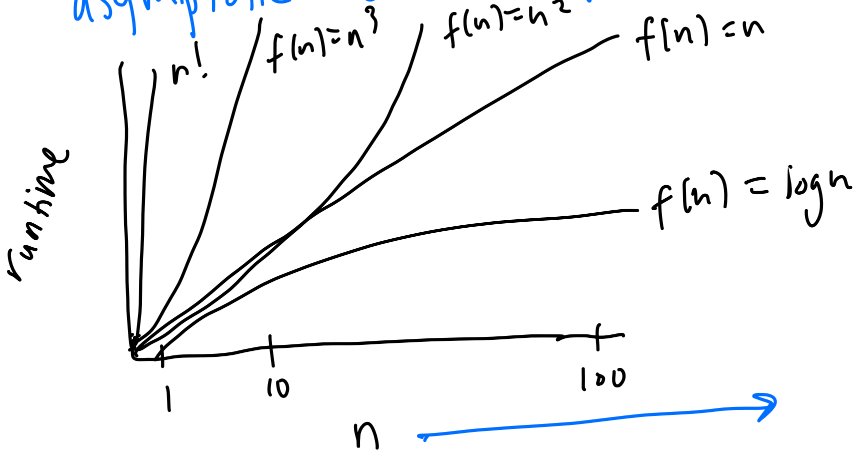
↓

# of constant-time operations (approximately)

I have a function/program that has runtime of $f(n)$.
You have a function/program that has runtime of $g(n)$.

which is better? → on large inputs.

asymptotic relationships



know these relationships:

$$1 \underbrace{\ll}_{\text{asymptotically smaller}} \log n \ll n \ll \underline{n \log n} \ll n^2 \ll n^3 \ldots\ldots \ll 2^n \ll 3^n \ll n!$$

$n^2 \log n$

asymptotically similar

see textbook

if $h(n)$ is a non-zero function, then

$$f(n) \ll g(n) \longrightarrow f(n)h(n) \ll g(n)h(n)$$

$h(n) = n$

$\log(n) << n \rightarrow n\log(n) << n^2$

faster

e.g.) $47n + 2n! + 17 >> 3^n + 102n^3$

dominant term method.

Big-O

$f(n)$ is $O(g(n))$ iff

there exist $c, k \in \mathbb{R}^+$ such that $0 \leq f(n) \leq c\,g(n)$ for every $n \geq k$.

we don't care about constant coefficients

e.g.) $f(n) = 3n^2 + 17n$

$g(n) = n^3$

I want to show $3n^2 + 17n$ is $O(n^3)$.

$c = 3$

$k = 100$

$0 \leq 3n^2 + 17n \leq 3n^3$

$\boxed{3n^2 + 17n \leq 3n^3}$

for $n \geq$ some k

$\boxed{n \geq 100}$

Proof by inequality induction

it is NOT the case that $3n^3$ is $O(n^2)$

e.g.) $2n^2$ is $O(n^2)$

$c = 4$

$2n^2 \leq 4n^2$ for $n \geq 1$

n^2 is $O(2n^2)$

$c = 1/2$

$n^2 \leq n^2$ for $n \geq 1$

then, n^2 is $\Theta(2n^2)$
$2n^2$ is $\Theta(n^2)$