20 Jan 2026

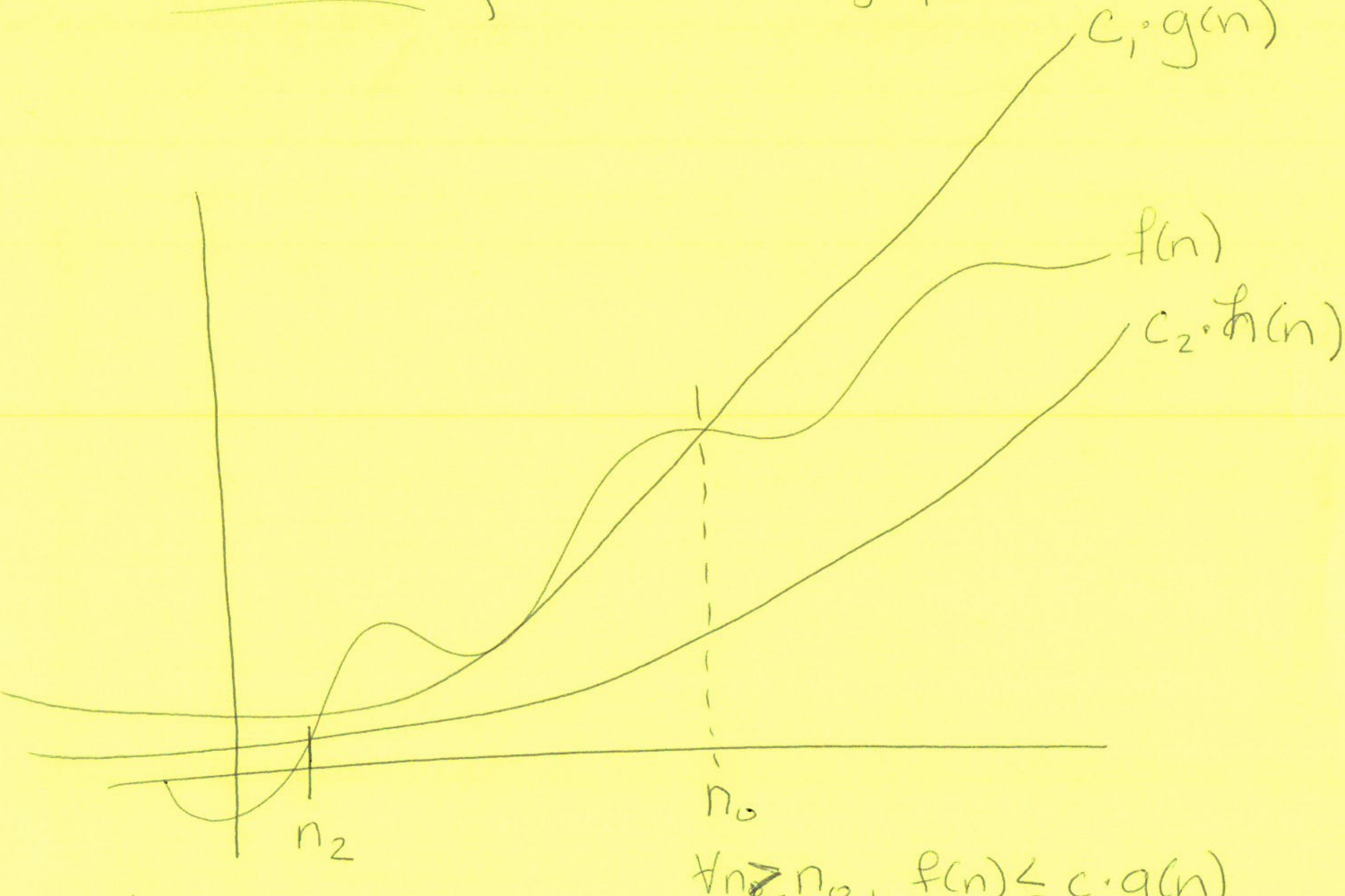
Asymptotics

worst-case
- runtime of binary search $\Theta(\log n)$
  "logrithmic time"

↑ "counting" how many steps
precise ans
maybe looks like

$7\log n + 32$
exact counting

$\Theta(\log n)$
asymptotic



$\forall n \geq n_2$,
$f(n) \geq c_2 \cdot h(n)$
$\Longleftrightarrow f(n)$ is $\Omega(h(n))$
$\Updownarrow$
$h(n)$ is $O(f(n)$

$\forall n \geq n_0,\ f(n) \leq c \cdot g(n)$
$\Longleftrightarrow f(n)$ is $O(g(n))$
$\Updownarrow$
$g(n)$ is $\Omega(f(n))$

What can f(n) represent?

- worst-case runtime
- space complexity
- average-case runtime (or expected runtime)

☆ asymptotics are not just for runtimes! ☆

$\mathbb{N} = \{0, 1, 2, \ldots\}$

$\mathbb{Z}_+ = \{1, 2, 3, \ldots\}$

Q: How do we prove $f(n)$ is $O(n^2)$?

need $\exists n_0 \in \mathbb{N}, c \in \mathbb{R}_+$ such that

$\boxed{\forall n \geq n_0,\ f(n) \leq cn^2.}$ *

Step 1: Find n_0 & c that work

Step 2: Prove * by induction

$f(n) = n^2 - 2$ is $O(n^2)$

$g(n) = n^2$

Try: $n_0 = 0$ $c = 1$

~~Base case: Prove $f(n_0)$ is $O($~~

Base case: Prove: ~~$f(n_0) \leq 1 \cdot n^2$~~

WTS: $f(n_0) \leq c \cdot g(n_0)$ ← what I want to show

$n_0^2 - 2 \leq 1 \cdot n_0^2$

$\quad\ \ \wedge$

$\quad n_0^2$

$\boxed{\text{Start}}$ here in scratch work.

$\boxed{\text{End}}$ here in a proof.

Proof: $f(n_0) = n_0^2 - 2$ by dfn ← Start w/ known things

$\leq n_0^2$ by adding

$= 1 \cdot n_0^2$

$= c \cdot g(n_0)$

$\therefore f(n_0) \leq c \cdot g(n_0)$ ← end w/ new stuff

$f(n) = n^2 + 2$ is $O(n^2) \rightarrow g(n) = n^2$

$\hookrightarrow$ use $n_0 = 1$ and $c = 3$

Every Proof

1. Claim / Set it up
2. Prove it
3. Conclude

Base case $\overset{\text{Prove}}{—}$ $f(n_0) \leq c \cdot g(n_0)$

Inductive assumption: Let $n \geq 1$,

assume $f(n) \leq c \cdot g(n)$.

$\iff n^2 + 2 \leq 3(n^2)$

here, can use n, k, $n-1$, $k-1$

Inductive step: We need to show that

$f(n+1) \leq c \cdot g(n+1)$

WTS

$(n+1)^2 + 2 \qquad 3 \cdot (n+1)^2$

$\underline{n^2} + 2n + 1 + \underline{2} \qquad \underline{3n^2} + 2n + 3$

$f(n) + 2n + 1 \qquad \underline{\underline{3g(n^2) + 3(2n+1)}}$

$\wedge\!|$ add any + #

$\underline{\underline{f(n) + 3(2n+1)}} \qquad \uparrow$ IA.

scratch-work / thinking

Proof: $f(n+1) = (n+1)^2 + 2$ by def of f

$= n^2 + 2n + 1 + 2$ by expanding

$= f(n) + 2n + 1$ by defn of f

$\leq f(n) + 3(2n+1)$ by adding a pos # $2(2n+1)$

$\leq 3 \cdot g(n^2) + 3(2n+1)$ by def of g

$\vdots$

$\leq c \cdot g(n+1)$, as desired $\boxplus$

Q5: $\Theta(n)$ defines an equiv. relation.

<u>Def'n</u> Equivalence relation: a binary relation $\sim$ on a set X that satisfies

1. reflexive $a \sim a \quad \forall\, a \in X$
2. symmetric $a \sim b \iff b \sim a$
3. transitive $a \sim b$ and $b \sim c \implies a \sim c$

X = the set of all fcns $\mathbb{N} \to \mathbb{R}$

e.g. $f(n) = n^2 - 2$ is $\Theta(n^2 - 2)$

$f(n) = n^2 + 3$ is $\Theta(n^2) \iff f(n) = n^2$ is $\Theta(n^2 + 3)$

$$\Theta(n^2) = \Theta(n^2 - 2) = \Theta(60n^2 + 10)$$

↑ preferred way to write it