Lecture 6 - January 23

Asymptotic Analysis of Algorithms

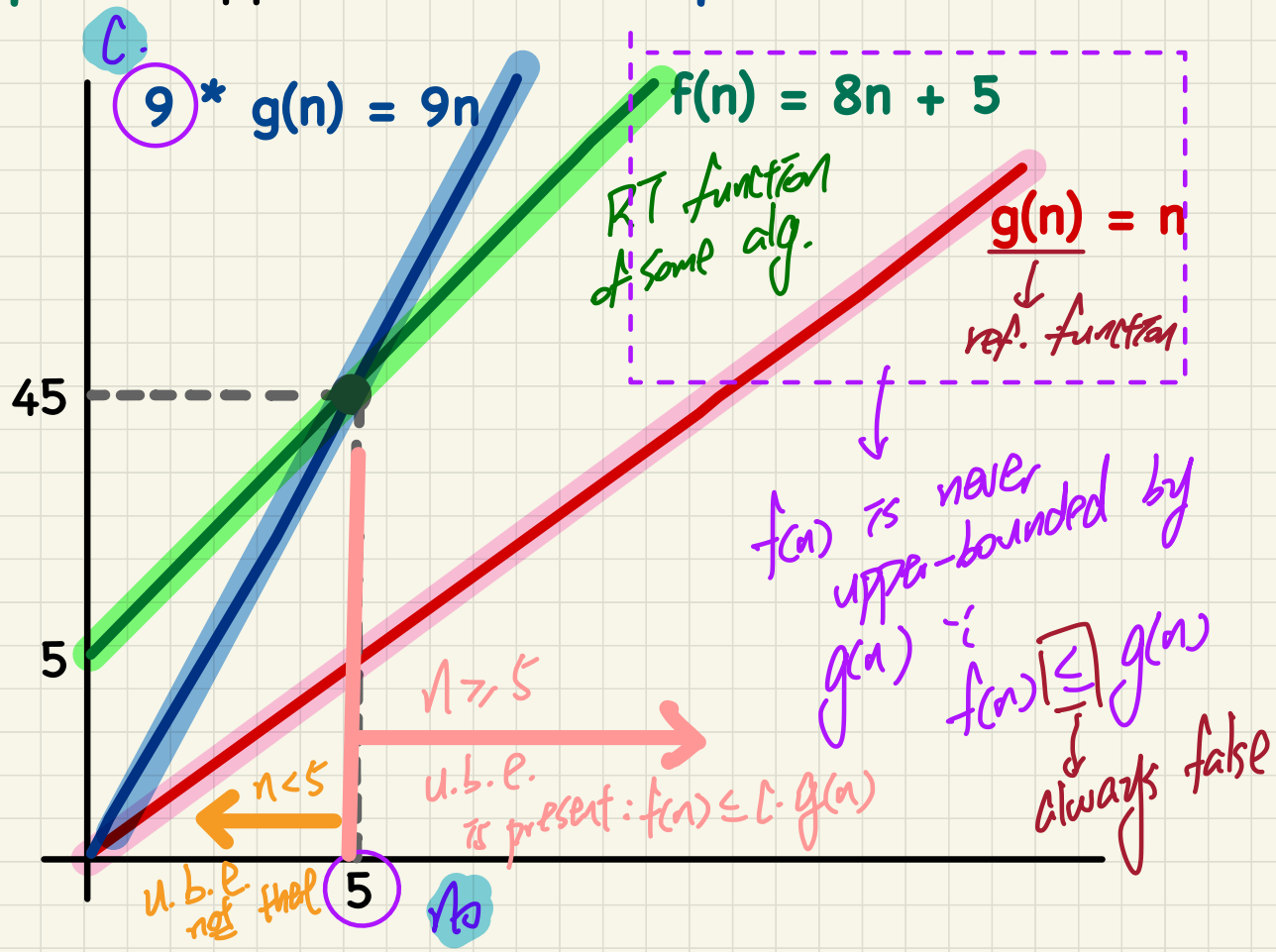
Big-O: Pred. Def., Properties, Examples
Correct vs. Accurate Asymptotic U.B.
Deriving U.B. from Code: Basic Examples

Announcements/Reminders

- **Assignment 1** due next Monday
- ***splitArrayHarder***: an **extended** version released
- Office Hours: 3pm to 4pm, Mon/Tue/Wed/Thu
- Contact Information of TAs on common eClass site

Asymptotic Upper Bound: Example

f(n) is order of O(g(n))



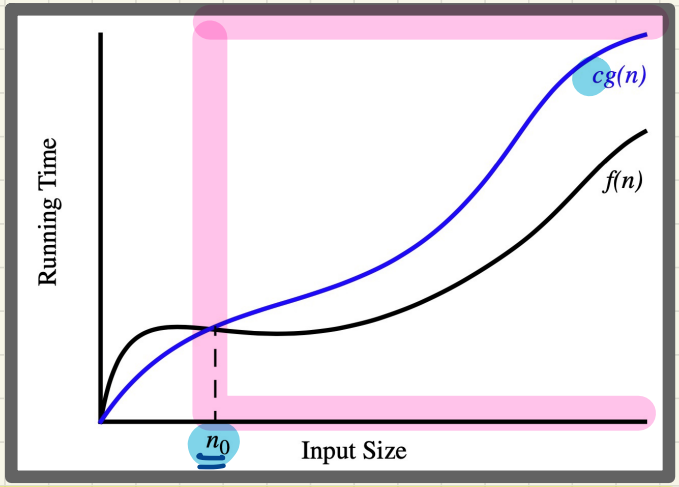
Asymptotic Upper Bound (Big-O): Alternative Formulation

Known:

$f(n) \in O(g(n))$ if there are:
- A real *constant* $c > 0$
- An integer *constant* $n_0 \geq 1$

such that:

$$f(n) \leq c \cdot g(n) \quad \text{for } n \geq n_0$$



Q. Formulate the definition of "f(n) is order of O(g(n))" using **logical** operator(s): $\neg, \wedge, \vee, \Rightarrow, \forall, \exists$

$$f(n) \in O(g(n)) \equiv \exists c, n_0 \cdot (c > 0 \wedge n_0 \geq 1 \wedge (\forall n \cdot n \geq n_0 \Rightarrow f(n) \leq c \cdot g(n)))$$

Proving f(n) is O(g(n))

If $f(n)$ is a polynomial of degree d, i.e.,

$$f(n) = a_0 \cdot n^0 + a_1 \cdot n^1 + \cdots + a_d \cdot n^d$$

and $a_0, a_1, \ldots, a_d$ are integers (i.e., negative, zero, or positive), then $f(n)$ is $O(n^d)$.

We prove by choosing

$$c = |a_0| + |a_1| + \cdots + |a_d|$$
$$n_0 = 1$$

Upper-bound effect: $n_0 = 1$? $[f(1) \leq (|a_0| + |a_1| + \cdots + |a_d|) \cdot 1^d]$

Upper-bound effect holds? $[f(n) \leq (|a_0| + |a_1| + \cdots + |a_d|) \cdot n^d]$

$* \; 1^0 = 1^1 = \cdots = 1^d = 1$

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

(1) $f(1) \leq c \cdot 1^d$

(2) $f(n) \leq c \cdot n^d$ $(n > 1)$

$** \; a_0 \leq |a_0|, \; a_1 \leq |a_1|, \; \ldots, \; a_d \leq |a_d|$

$f(1) = a_0 \cdot 1^0 + a_1 \cdot 1^1 + \cdots + a_d \cdot 1^d \overset{*}{=} (a_0 + a_1 + \cdots + a_d) \cdot 1^d \overset{**}{\leq} (|a_0| + |a_1| + \cdots + |a_d|) \cdot 1^d$

$*** \; n^0 \leq n^d, \; n^1 \leq n^d, \; \ldots, \; n^d \leq n^d$

$f(n) = a_0 \cdot n^0 + a_1 \cdot n^1 + \cdots + a_d \cdot n^d \overset{***}{\leq} (a_0 + a_1 + \cdots + a_d) \cdot n^d \overset{**}{\leq} (|a_0| + |a_1| + \cdots + |a_d|) \cdot n^d$ $(n > 1)$

Exercise: Prove $f(n) = 5n^4 - 3n^3 + 2n^2 - 4n + 1$ is $O(n^4)$.

$5 \cdot n^4 - 3n^3 + 2n^2 - 4n + 1 \cdot n^0$

(1) Guess: $O(n^4)$ ↓ $g(n)$

(2) Prove

choose $C = |5| + |(-3)| + |2| + |(-4)| + |1| = 15$

$n_0 = 1$

Verify:

$f(1) \leq 15 \cdot g(1)$

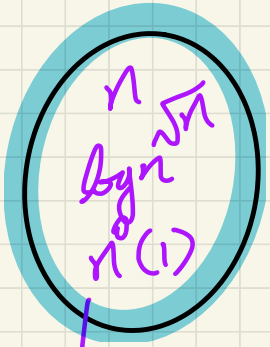
$5 - 3 + 2 - 4 + 1 \leq 15 \cdot 1$ ✓

Big-O Properties (1): Members in a Family

Each member f(n) in O(g(n)) is such that:

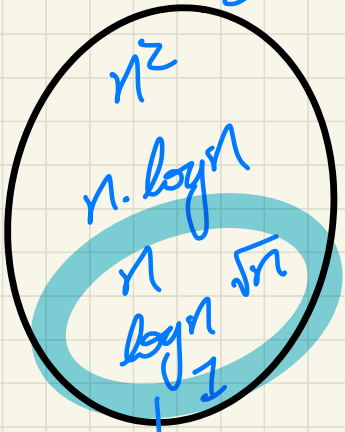
Higest Power of f(n) <= Highest Power of g(n)

O(n)



functions growing "no faster" than n should be included

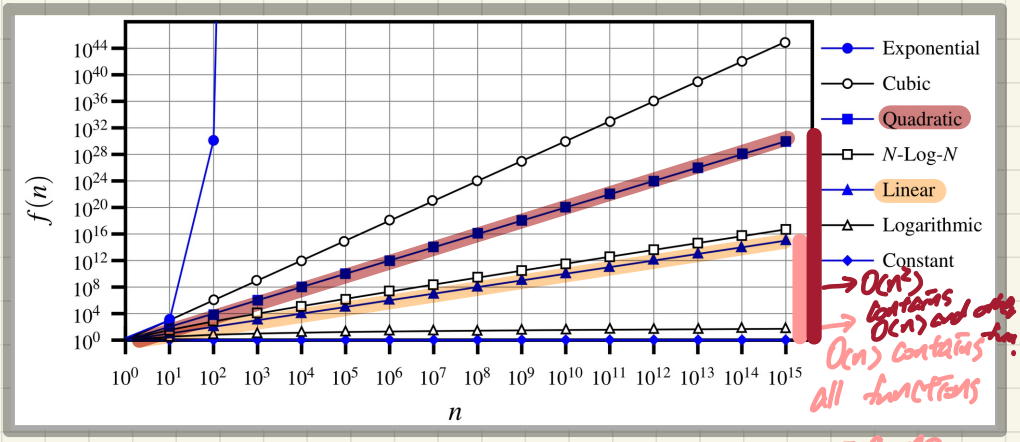
O(n^2)



O(n) is "contained" within $O(n^2)$

$2^n \notin O(n^2)$

Functions: Rates of Growth



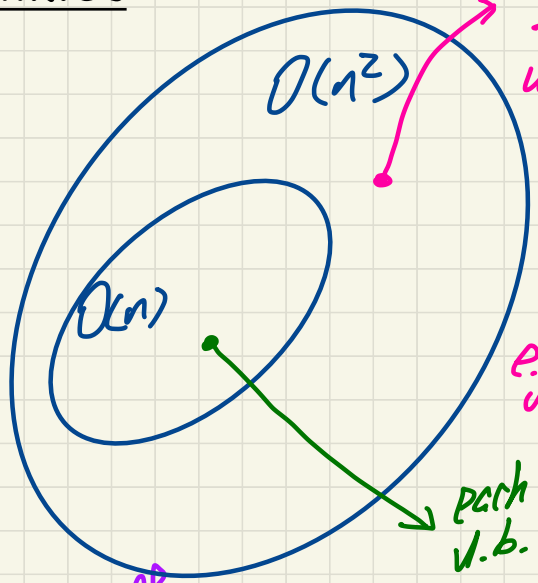
$O(n^2)$ contains $O(n)$ and others

$O(n)$ contains the all functions growing no faster than n

Big-O Properties (2): Relating Families

$O(n)$ $\subseteq$ $O(n^2)$

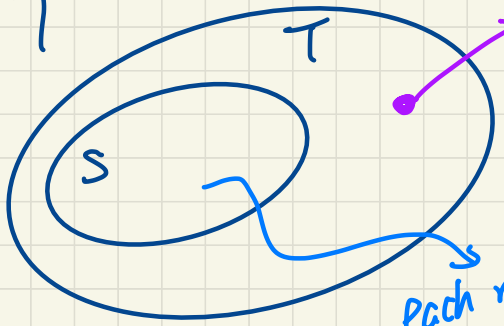
Correct
(1) $\subseteq$ → it's possible $O(n) = O(n^2)$
(2) $\subset$ correct & accurate
(3) $\supseteq$
(4) $\supset$



at least one function u.b. by n^2 but not u.b. by n.
e.g. n^2, $n \log n$

each function u.b. by n can also be u.b. by n^2

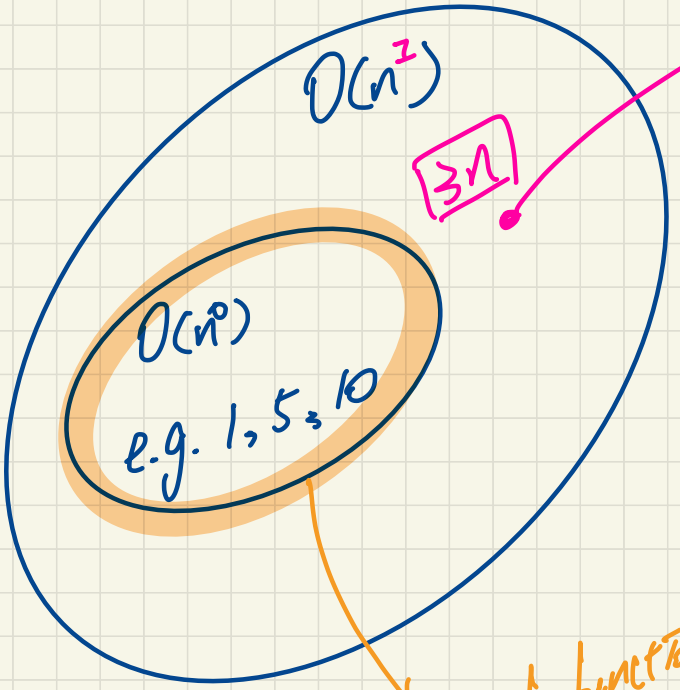
$S \subset T$



at least one member in T that's not a member in S.

each member in S is also a member in T

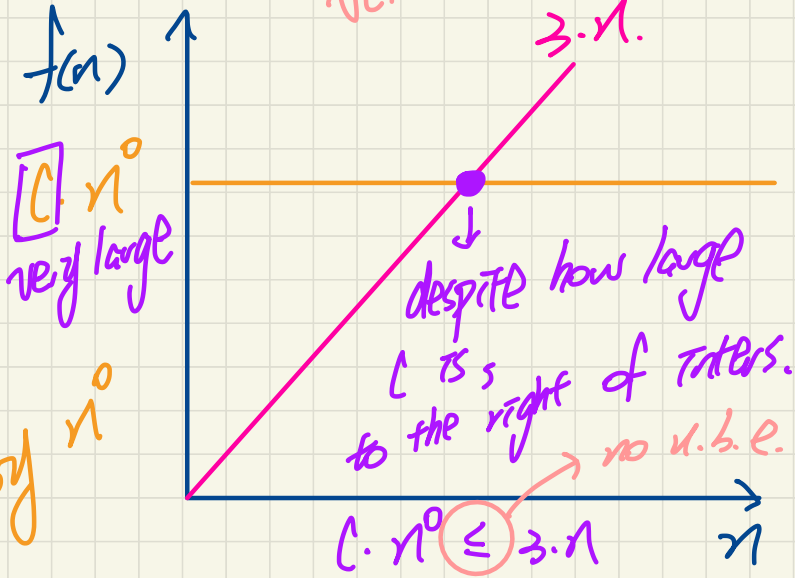
$O(n^0) \subset O(n)$



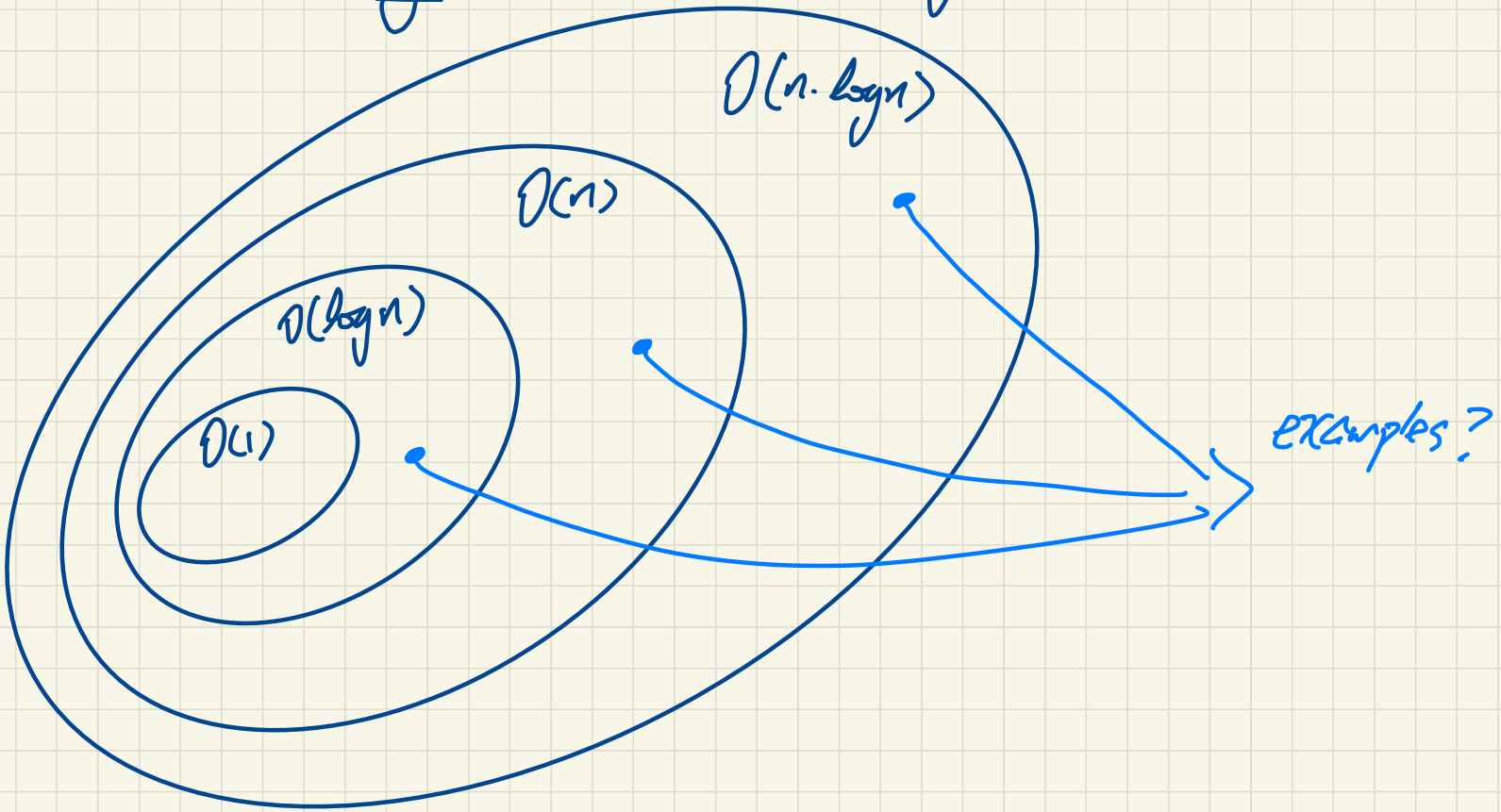
at least a function u.b. by n^1 but not u.b. by n^0

even with a VERY LARGE C chosen

set of functions that can be u.b. by n^0



$O(n^0) \subset O(\log n) \subset O(n) \subset O(n \cdot \log n) \subset \dots$



Big-O Properties (3): Deciding Correct & Accurate Bound

e.g. $f_1(n) = 7n - 2$

$f_2(n) = 4n^2 - 3n + 6$

$7n-2$ is order of ($\in$)

- $O(n)$ → most accurate (tightest U.b.)
- $O(n^2)$
- $O(n^3)$
- $O(2^n)$

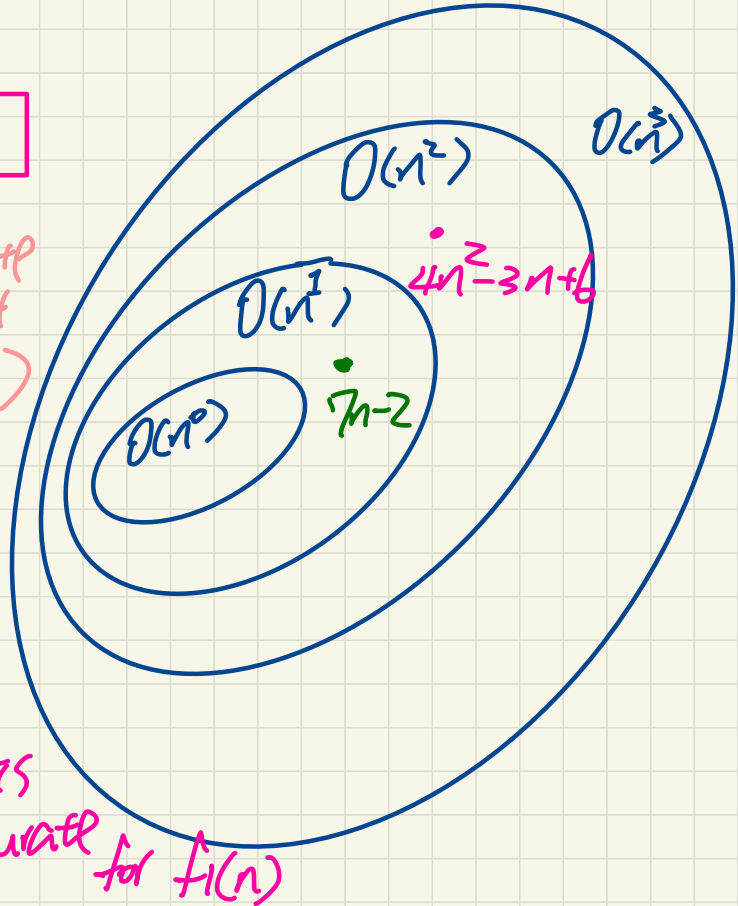
all correct

misleading.

$f_1(n)$ is $O(n^3)$ → $f_2(n)$ more efficient?

$f_2(n)$ is $O(n^2)$

No! ∵ the $O(n^3)$ is not accurate for $f_1(n)$



$O(7n^2 + 4n - 2)$ X not appropriate as the final answer

$O(n^2)$ ✓

Asymptotic Upper Bounds: Example (1)

Given $f(n) = 5n^2 + 3n \cdot \log n + 2n + 5$:

(1) What is f(n)'s **most accurate** asymptotic upper bound.

(2) **Prove** your claim.

(1) $O(n^2)$ — $g(n)$

(2) Choose $c = |5| + |3| + |2| + |5| = 15$

$n_0 = 1$

↳ verify $f(1) \leq 15 \cdot 1^2$

$5 \cdot 1^2 + 3 \cdot 1 \cdot \log 1 + 2 \cdot 1 + 5 \leq 15$ (true!)

($\log 1 = 0$, $2^0 = 1$)

Asymptotic Upper Bounds: Example (2) (Exercise)

Given $f(n) = 20n^3 + 10n \cdot \log n + 5$:

(1) What is f(n)'s **most accurate** asymptotic upper bound.

(2) **Prove** your claim.

Asymptotic Upper Bounds: Example (3)

Given $f(n) = 3 \cdot \log n + 2$:

(1) What is f(n)'s **most accurate** asymptotic upper bound.

(2) **Prove** your claim.

(1) $O(\log n)$

(2) choose $C = |3| + |2| = 5$

$n_0 = 1$ → verify: $f(1) \leq 5 \cdot \log 1$

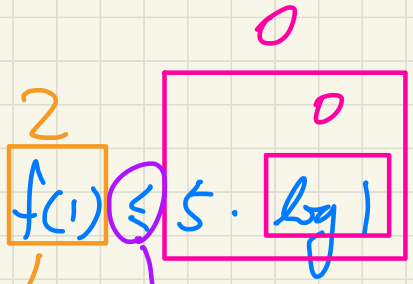
$f(1) = 3 \cdot \log 1 + 2$

fake

What if $n_0 = 2$

↳ $f(2) \leq 5 \cdot \log 2$

(Exercise) u.b.e?



Asymptotic Upper Bounds: Example (4) (Exercise)

Given $f(n) = 2^{n+2}$:

(1) What is f(n)'s **most accurate** asymptotic upper bound.

(2) **Prove** your claim.

Asymptotic Upper Bounds: Example (5)

(Exercise)

Given $f(n) = 2n + 100 \cdot \log n$:

(1) What is f(n)'s **most accurate** asymptotic upper bound.

(2) **Prove** your claim.