Lecture 5 - Monday, January 23

Announcements

- **Assignment 1** to be released tonight

  ↳ 6pm ~ 9pm

Lecture

Asymptotic Analysis of Algorithms

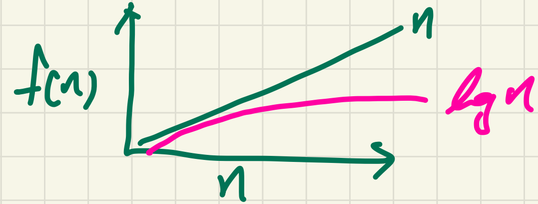
Asymptotic Upper Bound

n vs. $7n$

↳ asymptotically, they're just the family of "n" $\underline{\underline{O(n)}}$

$7n + 2n \cdot \log n + 3n^2$

input size



approximate the above running time function

lowest term

$n^{1+<1} = n^{1.\ldots}$

lower power from 1.

$7 \cdot n^1 + 2 \cdot \boxed{n^1 \cdot \log n} + 3 \cdot \boxed{n^2}$

this is what matters; disregarding $\underline{\underline{\text{ALL}}}$ lower terms

multiplicative constants

lower term

highest power

RT

$$\boxed{f(n) = 5}$$

findMax

↳ $7n - 2$

↳ (1) # POs

e.g. $n = 1 \rightarrow 5$ POs

$n = 10 \rightarrow 65$ POs

(2) relative RT

$f(0) = 5$

$f(10) = 5$

$f(1M) = 5$

→ RT is independendant of the input size

Polynomial: n^d

$d \geq 2$

$\cdot f(n)$ → $O(g(n))$

$f(n) \in O(g(n))$

- $f(n)$: RT function
  - ↳ inputsize → relative RT
  - e.g. findmax has relative RT: $7n-2$
- $g(n)$: reference function
  - (further manipulation on $g(n)$ expection)

$O(n)$ ✓

$O(\underline{\underline{2}} \cdot n)$ ✗

$O(\underline{\underline{2}}n + \underline{\tfrac{1}{2}})$ ✗

<u>Goal</u> Prove $f(n)$ is $O(\boxed{g(n)})$

↳ not including
(1) lower terms
(2) multiplicative constants

Asymptotic Upper Bound: Big-O

$O(g(n))$

$f(n) \in O(g(n))$ if there are:

- A real constant $c > 0$
- An integer constant $n_0 \geq 1$

such that:

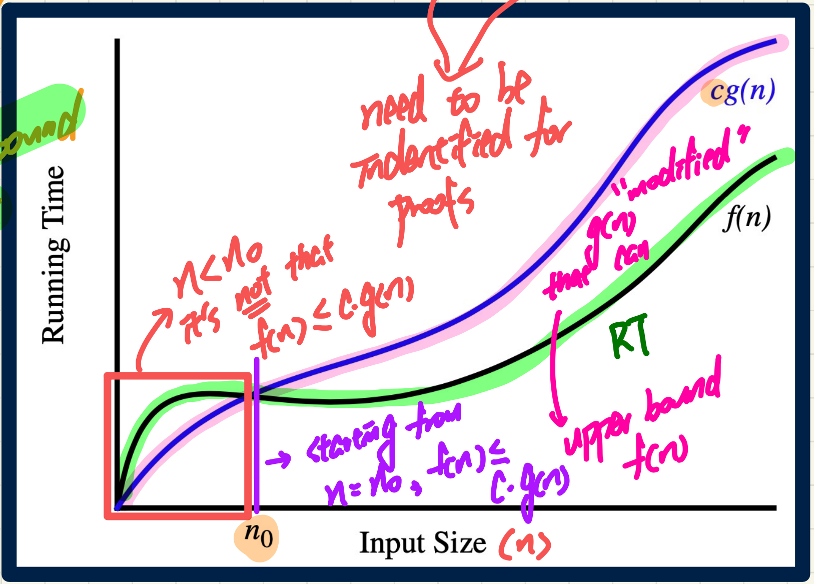
$$f(n) \leq c \cdot g(n) \quad \text{for } n \geq n_0$$

u.b.e.

multiplicative constant applied to g(n) to change its slope

f(n)

starting point of upper bound effect



Example:

$f(n) = 8n + 5$

$g(n) = n$ ref. function

Prove:

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

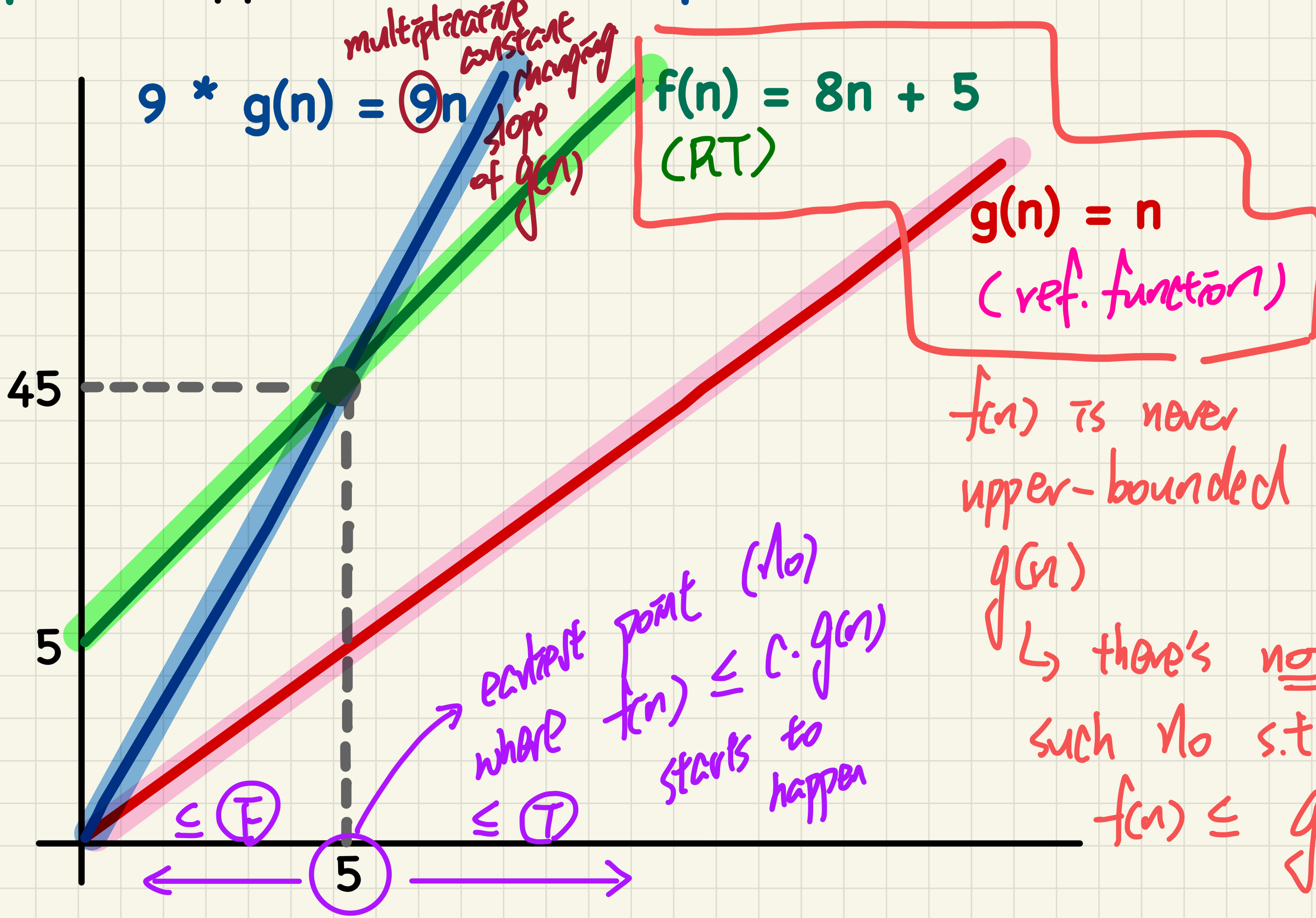
Choose:

$c = 9$

What about n0?

$f(n)$ is $O(\,g(n)\,)$

Asymptotic Upper Bound: Example



$f(n)$ is never upper-bounded by $g(n)$

↳ there's not such n_0 s.t. $f(n) \leq g(n)$

RT

higest power

$$f(n) = 5n^4 + 3n^3 + 2n^2 + 4n + 1 \cdot n^0$$

(1) Guess: $f(n)$ is $O(n^4)$

(2) Prove:

choose c: $|5| + |3| + |2| + |4| + |1| = 15$

choose n_0: 1.

Lecture

Asymptotic Analysis of Algorithms

Asymptotic Upper Bounds of Math Functions

Asymptotic Upper Bounds: Example (1)

$\log 1 = 0$

$$5n^2 + 3n \cdot \log n + 2n + 5 \text{ is } O([redacted])$$

Problem: (1) State and (2) prove the asymptotic upper bound of the above function.

(1) $O(n^2)$

(2) Prove by choosing:

$C = |5| + |3| + |2| + |5| = 15$

$n_0 = 1$

Verify: show $f(n) \le 15 \cdot n^2$ when $n = 1$

$5 \cdot 1^2 + 3 \cdot 1 \cdot \log 1 + 2 \cdot 1 + 5 = 12 \le 15 \cdot 1^2$