cse 101 11-3-20

PA3: ext. 2 days

Asymptotic growth of fcns

Analogy: comparing fcn. growth rates vs. comparing numbers.

$f(n) = O(g(n)) \longleftrightarrow x \leq y$

$f(n) = \Omega(g(n)) \longleftrightarrow x \geq y$

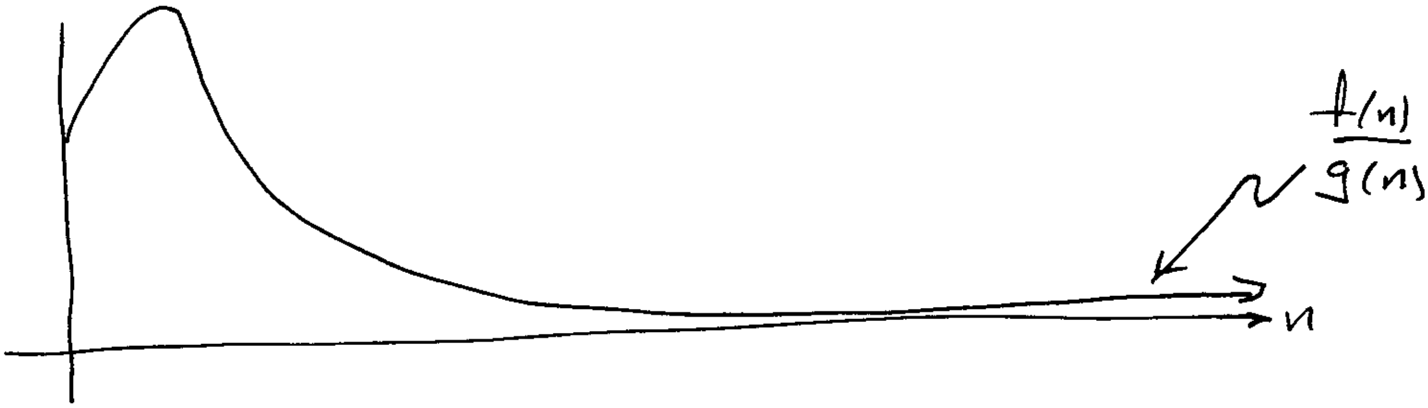
$f(n) = \Theta(g(n)) \longleftrightarrow x = y$

$f(n) = o(g(n)) \longleftrightarrow x < y$

$f(n) = \omega(g(n)) \longleftrightarrow x > y$

Defn

$$f(n) = o(g(n)) \text{ iff } \lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = 0$$

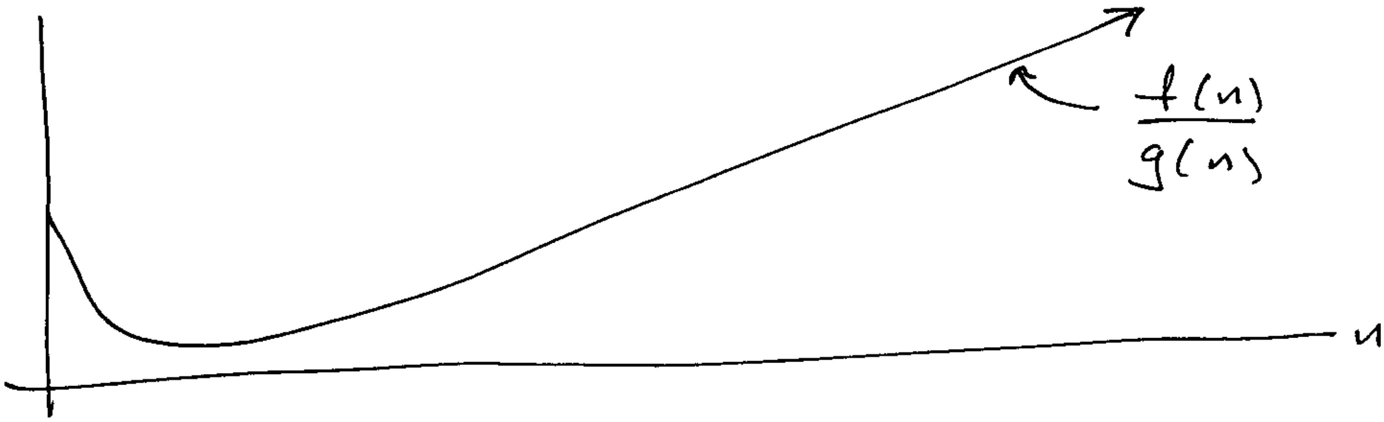


Thm: $f(n) = o(g(n)) \Rightarrow f(n) = O(g(n))$

note: converse $\Leftarrow$ is false!

Defn

$$f(n) = \omega(g(n)) \text{ iff } \lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \infty$$



<u>Thm</u> $f(n) = \omega(g(n)) \Rightarrow f(n) = \Omega(g(n))$

<u>note</u>: converse is false: $\not\Leftarrow$

<u>Thm</u>

$f(n) = o(g(n))$ iff $g(n) = \omega(f(n))$

<u>Ex</u> $\ln(n) = o(n)$ why?

$$\lim_{n\to\infty} \left(\frac{\ln(n)}{n}\right) = \lim_{n\to\infty} \left(\frac{1/n}{1}\right) = 0 \quad \checkmark$$

l'Hopitals rule

<u>also</u>:

$\log_b(n) = o(n)$ for any $b > 1$

<u>Thm</u>: if $a > 1$, $b > 1$ then

$$\log_a(n) = \Theta(\log_b(n))$$

Ex $\ln(n) = o(n^{\alpha})$ for $\alpha > 0$ ✓

$$\lim_{n\to\infty}\left(\frac{\ln(n)}{n^{\alpha}}\right) = \lim_{n\to\infty}\left(\frac{1/n}{\alpha\cdot n^{\alpha-1}}\right)$$

$$= \lim_{n\to\infty}\left(\frac{1}{\alpha\cdot n^{\alpha}}\right) = 0$$

Ex. $(\ln(n))^{k} = o(n^{\alpha})$ for $\alpha>0,\ k>0$

Exercise

Ex. $n^{k} = o(e^{n})$ for any $k>0$

$$\lim_{n\to\infty}\left(\frac{n^{k}}{e^{n}}\right) = \lim_{n\to\infty}\left(\frac{k n^{k-1}}{e^{n}}\right)$$

$\vdots$ k applications of l'HoP.

$= 0$

Ex. $n^{k} = o(b^{n})$ for any $k>0,\ b>1$.

Exercise

Ex.

$$a^n = \begin{cases} o(b^n) & \text{iff } a < b \\ \Theta(b^n) & \text{iff } a = b \\ \omega(b^n) & \text{iff } a > b \end{cases}$$

$$\frac{a^n}{b^n} = \left(\frac{a}{b}\right)^n \longrightarrow \begin{cases} 0 & \text{iff. } a < b \\ 1 & \text{"} \quad a = b \\ \infty & \text{"} \quad a > b \end{cases}$$

Heirarchy of growth rates

	$\vdots$	$\uparrow$
	4^n	
exp. fcns	3^n	intractable algorithms
	e^n	
	2^n	
	$(3/2)^n$	
	$\vdots$	$\downarrow$

	$\vdots$	$\uparrow$
	n^4	
Poly fcns	n^3	Practical algorithms
	$n^{2.5}$	
	n^2	
	n	
	$\sqrt{n} = n^{1/2}$	
	$\vdots$	

	$\vdots$	
log fcns	$(\log(n))^3$	
	$(\log(n))^2$	$\downarrow$
	$\vdots$	

Efficiency of BFS

Let $n = |V(G)|$, $m = |E(G)|$

- Initialization cost (lines 1-9)

  $\Theta(n)$

- Queue ops cost

  $\Theta(n)$ since each vertex enters queue at most once.

- Scanning adj lists cost:

  $\Theta(m)$ since total length of all adj lists

  $$= \begin{cases} m & \text{digraph} \\ 2m & \text{graph} \end{cases}$$

- Total cost of BFS $= \Theta(n+m)$

  Linear in # of bytes needed to store adj list representation.

Depth First Search (DFS)

- has no source
- discovers all vertices in G.
- requires vertex attributes:

  - $color[x]$: (w, g, b)
  - $P[x]$: Parent of x in Pred. subgraph
  - $d[x]$: discover time of x
  - $f[x]$: finish " " "

- Calls recursive subroutine Visit(x), which discovers & finishes x.

- uses variable time, int

$$0 \leq time \leq 2n \quad (n = |V|)$$

time is static over all rec. calls to Visit.

- see pseudo-code
  - handout
  - textbook
  - webpage