CSE 101 2-4-21

- Pa4: ext. 2 days (last & only one)
- Quiz3 Tue 2-9-21

Pa4 stuff: cleanUp()

Ex. ③ ≠ ② ① X | ④ ≠ ≠ ≠ X X X X ≠ (5)

cleanUp()

3 2 1 | 4 (3)

Ex. ② ≠ ③ ① | X ≠ X ≠

2 3 1 |

Ex. ② ① x | x ③ ≯ x ≮ ≯

↓

2 1 | 3

Ex ① ② x | x ③ ≯ ≮ x

↓

1 2 | 3

Continue Algorithm Runtime ...

exercises

- show $n^{\alpha} = \begin{cases} o(n^{\beta}) & \alpha < \beta \\ \Theta(n^{\beta}) & \alpha = \beta \\ \omega(n^{\beta}) & \alpha > \beta \end{cases}$ ✓ ✓ ✓

$$\frac{n^{\alpha}}{n^{\beta}} = n^{\alpha-\beta} \longrightarrow \begin{cases} 0 & \alpha < \beta \\ 1 & \alpha = \beta \\ \infty & \alpha > \beta \end{cases}$$

note: $\frac{f(n)}{f(n)} = 1 \quad \therefore \quad 1 \le \frac{f(n)}{f(n)} \le 1 \quad (\text{all } n)$

$$\therefore f(n) = \Theta(f(n))$$

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

- let $c > 0$, then

$$c \cdot f(n) = O(f(n))$$
$$c \cdot f(n) = \Omega(f(n))$$
$$c \cdot f(n) = \Theta(f(n)).$$

- $\log_b(n) = \Theta(\log_a(n))$ for any $a, b > 1$

Hint: $\frac{\log_b(n)}{\log_a(n)} = \text{constant}$

- If $\lim_{n \to \infty} \left(\frac{f(n)}{g(n)} \right) = L$, then

a. $L = 0 \iff f(n) = o(g(n))$ ✓

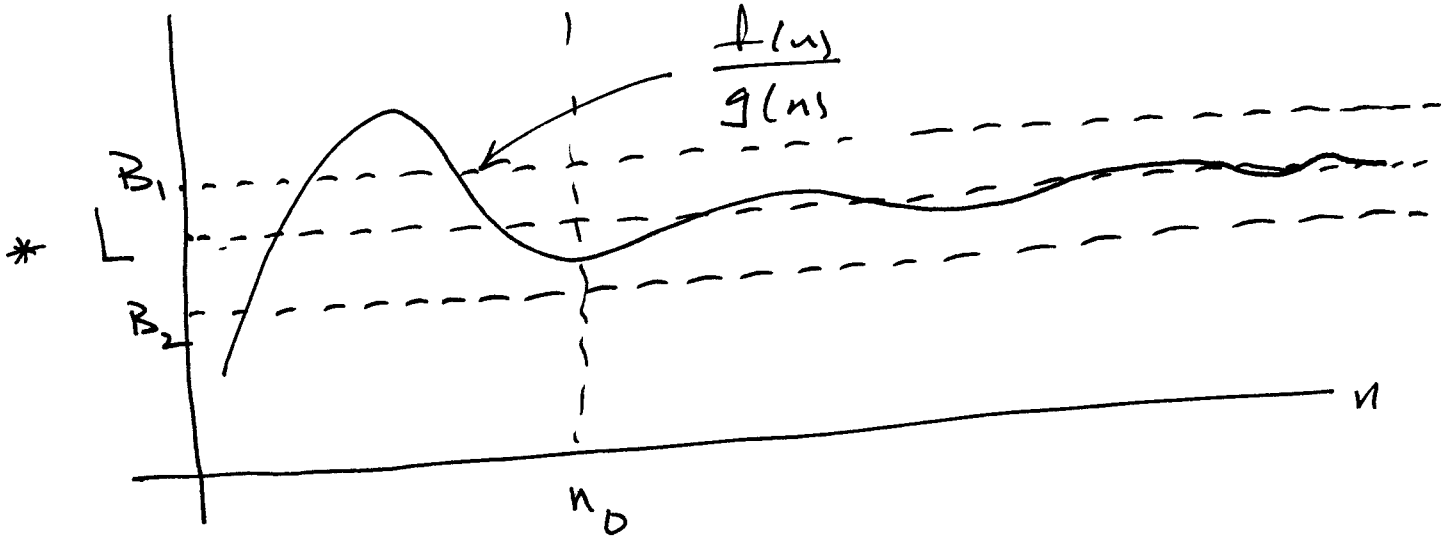
b. $0 \le L < \infty \Rightarrow f(n) = O(g(n))$

* → c. $0 < L < \infty \Rightarrow f(n) = \Theta(g(n))$

d. $0 < L \le \infty \Rightarrow f(n) = \Omega(g(n))$

⇐ is false

e. $L = \infty \iff f(n) = \omega(g(n))$ ✓



$0 < L < \infty$

- Rank the following functions from lowest to highest growth rate.

$$\left\{ n^2,\ \ln(n),\ (\ln(n))^2,\ \ln(n^2),\ n\ln(n),\ \sqrt{n},\ n\sqrt{n},\ \ln(\ln(\sqrt{n})),\ 2^{\ln(n)},\ 2^n,\ 2^{3^n},\ 3^{2^n} \right\}$$

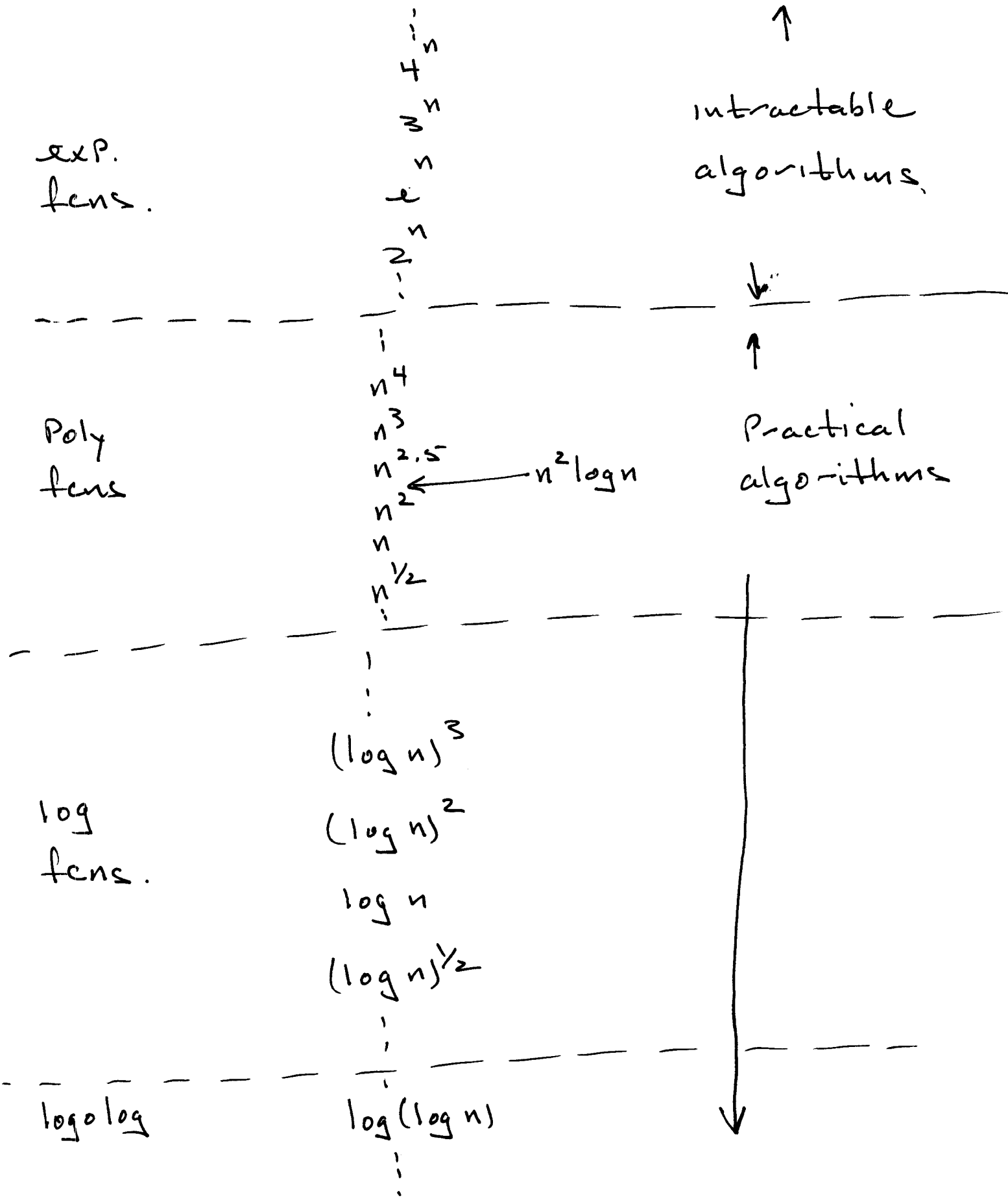
$$\ln(\ln(\sqrt{n})) = \ln(\ln(n^{1/2})) = \ln(\tfrac{1}{2}\ln n)$$

$$= \ln\tfrac{1}{2} + \ln(\ln(n)) = \ominus(\ln(\ln n))$$

$$\frac{\ln(\ln(n))}{\ln(n)} \to \frac{1/\ln n \cdot 1/n}{1/n} = \frac{1}{\ln n} \to 0$$

$\therefore \ln(\ln n) = o(\ln n)$ $\therefore$ $\boxed{\ln(\ln\sqrt{n}) = o(\ln(n))}$

Hierarchy of growth rates



SelectionSort(A) $A = (A_1, A_2, \ldots, A_n)$

1. $n = length[A]$
2. for $i = 1$ to $n-1$
3. $\quad imin = i$
4. $\quad$ for $j = i+1$ to n
5. $\quad\quad$ if $A[j] < A[imin]$ ← basic op.
6. $\quad\quad\quad imin = j$
7. $\quad A[i] \leftrightarrow A[imin]$ // swap

$\underbrace{A_1 \ldots\ldots A_{i-1}}_{\text{sorted}}$ $\underbrace{A_i\, A_{i+1} \ldots\ldots A_n}_{\text{unsorted}}$

$(i-1)$ smallest elements.

Basic operation: comparison of array elements. (line 5)

$$\#\text{comp} = (n-1)+(n-2)+(n-3)+\cdots+3+2+1$$

$$= \frac{n(n-1)}{2} = \frac{1}{2}n^2 - \frac{1}{2}n = \Theta(n^2)$$

BFS Runtime

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

- Initialization : $\Theta(n)$
- Enqueues & Dequeues : $O(n)$
- cost of inner for-loop : $\Theta(m)$
- total cost $= \Theta(n+m)$

$= \Theta(\#$ bytes needed to store adj list rep.$)$

DFS Runtime

let $n = |V(G)|$, $m = |E(G)|$.

- initialization : $\Theta(n)$
- main loop (minus visit()) : $\Theta(n)$
- for loop in visit (minus call visit()) : $\Theta(m)$
- # calls to visit : $\Theta(n)$
- total cost : $\Theta(n+m)$

$= \Theta($# bytes in adj. list rep.$)$

New ADT: Dictionary

- A state is a set of Pairs

  (key, value)

  $\{ (k_1, v_1), (k_2, v_2), \ldots\ldots, (k_n, v_n) \}$

- operations:
  - insert (key, value) } manipulation procedures
  - delete (key) } manipulation procedures
  - lookup (key) } access function

Implementations
- BST (chap. 12)
- RBT (chap 13)
- Hash Table (---)