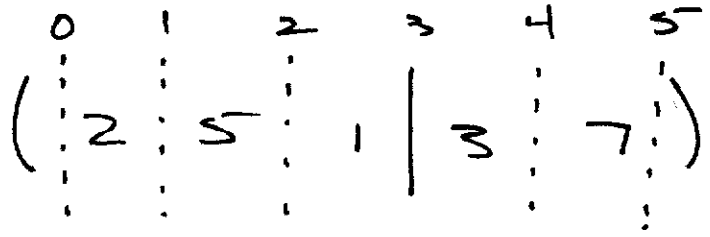
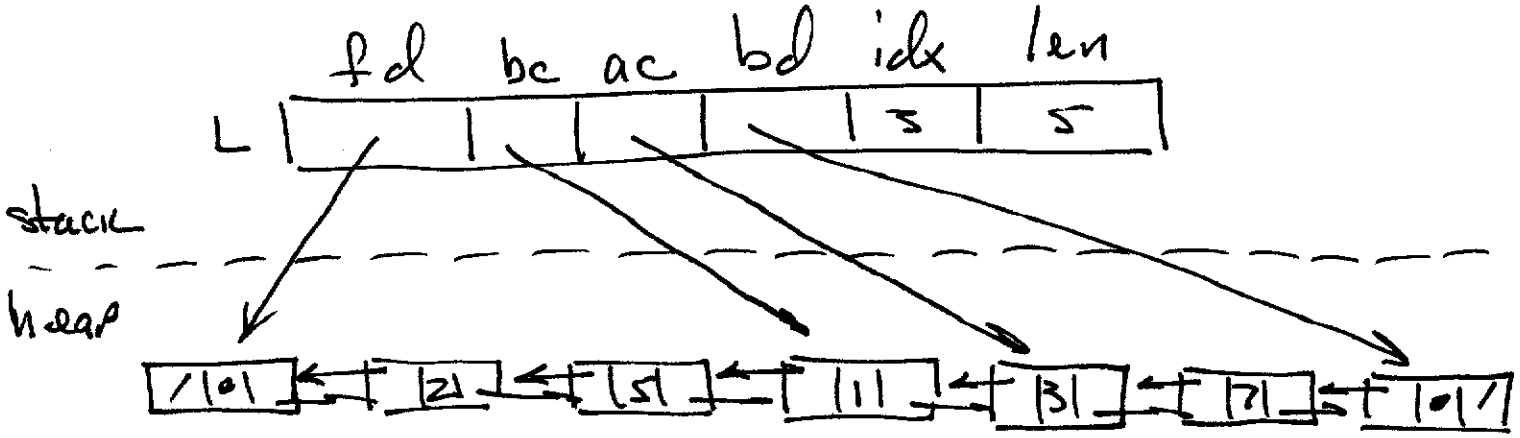
CSE 101 4-29-21

Part still:

client view:



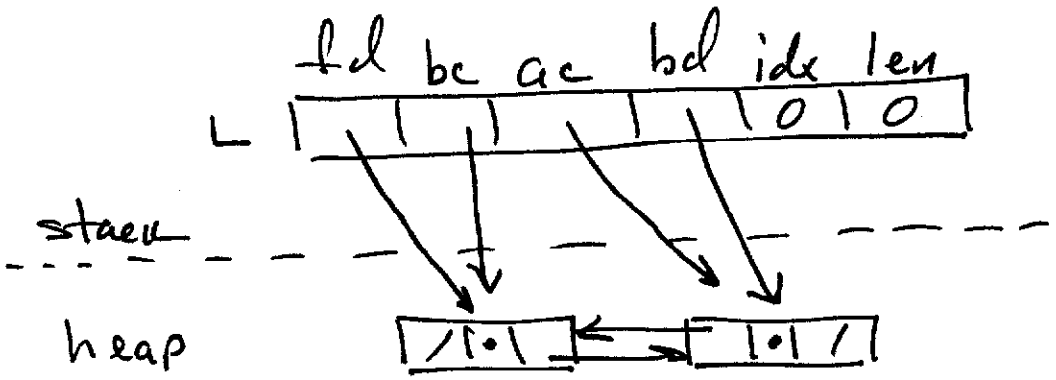
inside view



empty state:

client: (|)

inside:



const. & copy const. & dest.

see: Queue ex.
Stack ex.

-> operator

Node * N;

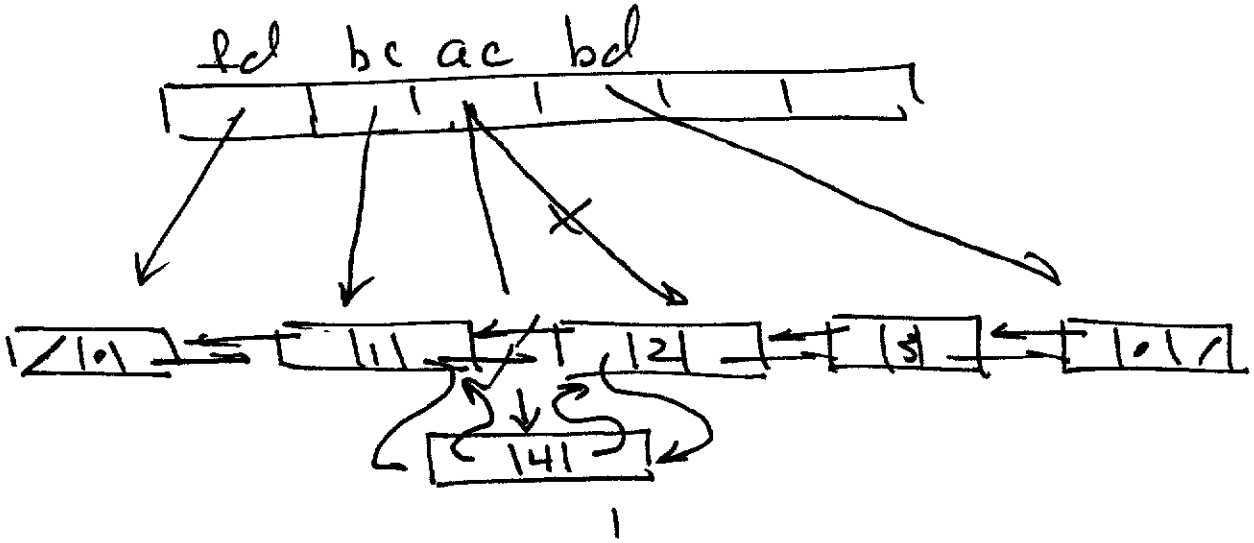
~~*N.next~~

(*N).next equiv to N->next

insertAfter()

client: (1 | 2 3)

inside:



insertAfter(4): (1 | 4 2 3)

Continue asymp. growth handout

Exercises

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

Proof:

$$\frac{n^\alpha}{n^\beta} = n^{\alpha-\beta} \xrightarrow[n\to\infty]{\text{as}} \begin{cases} 0 & \text{if } \alpha < \beta \\ 1 & \text{if } \alpha = \beta \\ \infty & \text{if } \alpha > \beta \end{cases}$$

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

- $cf(n) = O(f(n))$

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

  $\quad\quad = \Omega(f(n))$

  for any $c > 0$

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

  for any $a > 1$, $b > 1$.

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

  a. $L = 0 \iff f(n) = o(g(n))$

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

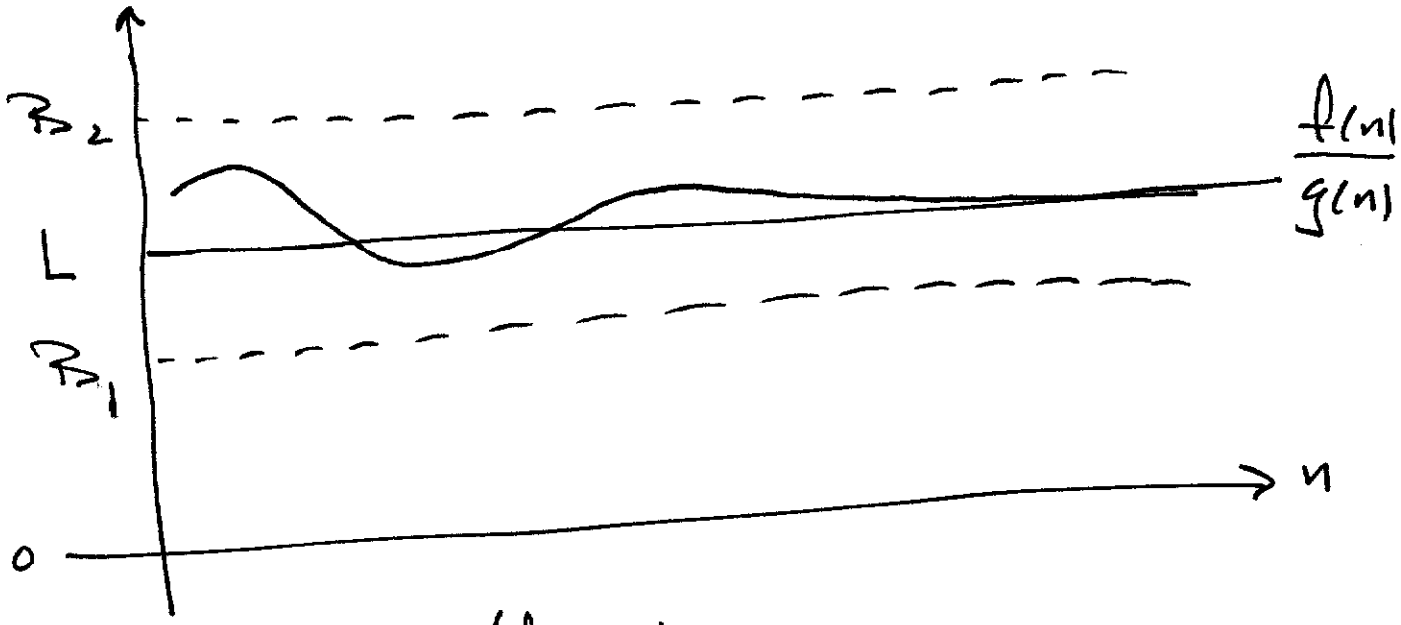
  c. $0 < L < \infty \implies f(n) = \Theta(g(n))$

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

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

  (b, c, d: $\not\Longleftarrow$)

look at case c:



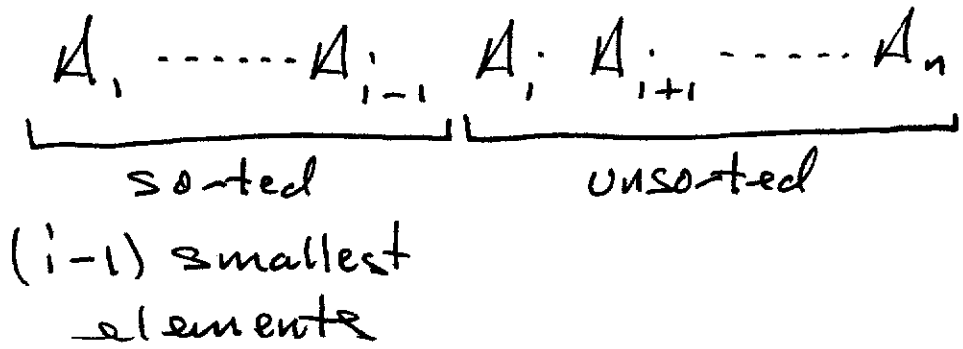
$$\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = L$$

- rank the following functions:

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

SelectionSort(A)

1. n = length[A]
2. for i = 1 to n-1
3. imin = i
4. for j = i+1 to n
5. if (A[j] < A[imin]) ← basic OP.
6. imin = j
7. A[i] ↔ A[imin] // swap



Basic operation in sorting Algorithm:

Comparison of array elements

(# comparisons as a fcn of n)

	# comp
i = 1	n-1
i = 2	n-2
i = 3	n-3
⋮	
i = n-1	1

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

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

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

best, worst and avg !!

next time:

RFS, DFS runtimes.