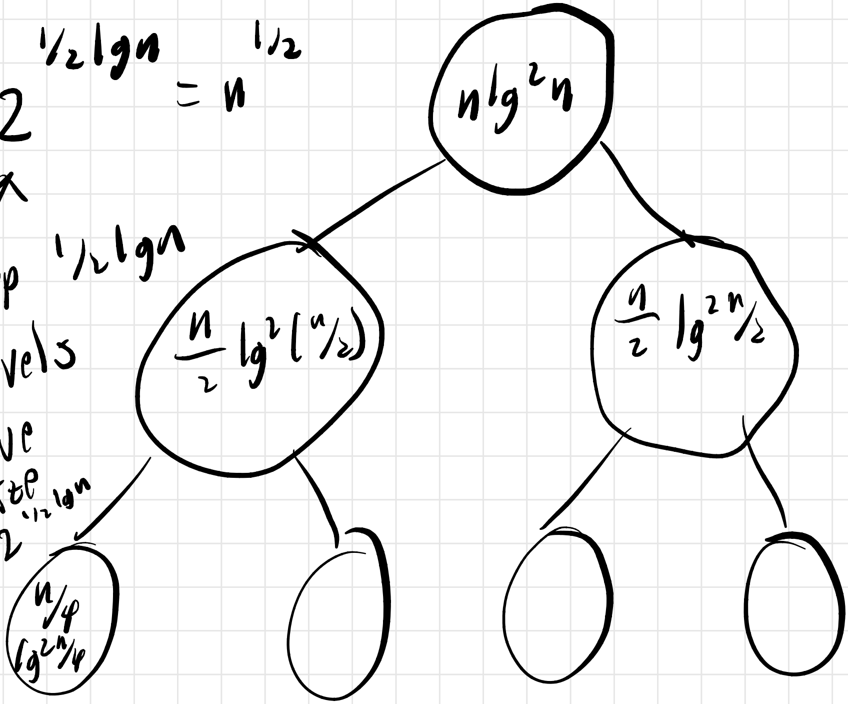
$\log(n^{1/2}) = \frac{1}{2}\log n$

$E(n) = 2E(n/2) + n\lg^2 n = \Theta(n\log^3 n)$

$2^{\frac{1}{2}\lg n} = n^{1/2}$

top $\frac{1}{2}\lg n$ levels have size $\geq 2^{\frac{1}{2}\lg n}$



$n\lg^2 n$

$n\lg^2(n/2)$

$n\lg^2(n/4)$

$E(n) \leq \lg n \cdot (n\lg^2 n) = O(n\log^3 n)$

$E(n) \geq \frac{1}{2}\lg n \cdot (n\lg^2 n^{1/2})$

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

$= \Omega(n\log^3 n)$

$\log a + \log b = \log(mn) \leq \log((m+n)^2)$

$= 2\log(m+n)$

$= O(\log(m+n))$

$A[1..m]$, $B[1..n]$.

sorted

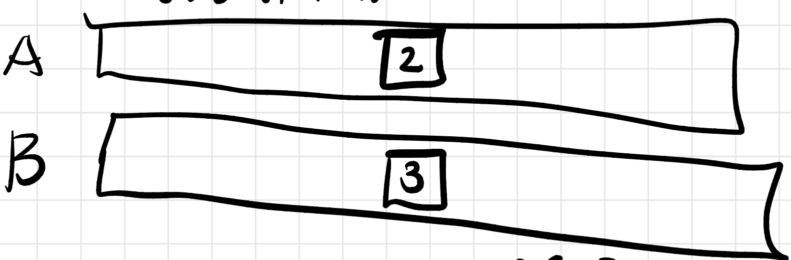
Find element of rank k in

$A \cup B$

if A

```
SELECTSORTED(A[1 .. m], B[1 .. n], k):
  if m = 1
      if k ≠ 1 and A[1] < B[k−1]
          return B[k−1]
      else if k > n or A[1] < B[k]
          return A[1]
      else
          return B[k]
  else if n = 1
      if k ≠ 1 and B[1] < A[k−1]
          return A[k−1]
      else if k > m or B[1] < A[k]
          return B[1]
      else
          return A[k]
  else if A[⌊m/2⌋] < B[⌊n/2⌋]
      if k ≤ ⌊m/2⌋ + ⌈n/2⌉
          return SELECTSORTED(A[1 .. m], B[1 .. ⌈n/2⌉], k)
      else
          return SELECTSORTED(A[⌊m/2⌋+1 .. m], B[1 .. n], k − ⌊m/2⌋)
  else
      if k ≤ ⌈m/2⌉ + ⌊n/2⌋
          return SELECTSORTED(A[1 .. ⌈m/2⌉], B[1 .. n], k)
      else
          return SELECTSORTED(A[1 .. m], B[⌊n/2⌋+1 .. n], k − ⌊n/2⌋)
```

$A[i]$ s.t. $i \le \lfloor m/2 \rfloor$



$B[i]$; $i \ge \lceil n/2 \rceil + 1$

these are > than at least

$\lfloor m/2 \rfloor + \lceil n/2 \rceil$ elements,

so if $k \le \lfloor m/2 \rfloor + \lceil n/2 \rceil$,
~~we~~ none of those are the answer.

$A[i]$ s.t. $i \le \lfloor m/2 \rfloor$ smaller than $\lceil m/2 \rceil + \lceil n/2 \rceil + 1$ elements,
so they have rank $\le m + n$

$$- (\lceil m/2 \rceil + \lceil n/2 \rceil + 1)$$

$$\le \lfloor m/2 \rfloor + \lceil n/2 \rceil$$

if $k > \lfloor m/2 \rfloor + \lceil n/2 \rceil$,
they are not the answer

$$WTF(i,j) = \begin{cases} 0 & \text{if } i \le 0 \text{ or } j \le 0 \\ X[j] + WTF(i-1,j) + WTF(i,\lfloor j/2 \rfloor) & \text{otherwise} \end{cases}$$

Time to eval $WTF(n,n)$?
using DP

$0 \le i \le n$
$0 \le j \le n$ $\Rightarrow$ so $O(n^2)$ subproblems

$O(1)$ time to eval each

so $O(1) \cdot O(n^2) = O(n^2)$ total

2.5/2.5 points

actually...
j takes on $O(\log n)$ values
so $O(n \log n)$ subproblems
$\Rightarrow O(n \log n)$ time

+2 EC

$$x[i] + \cancel{WTF(i-1, j)}$$

$$+ WTF(i, \lfloor j/2 \rfloor)$$

$$+ WTF(\lfloor i/2 \rfloor, j)$$

still $O(n \log n)$

now $O(\log^2 n)$

$$minCost(i,j) = \begin{cases} 0 & \text{if } i = 0 \text{ and } j = 0 \\ \infty & \text{if } i > 0 \text{ and } j = 0 \\ minCost(i, j-1) & \text{if } X[i] \neq Y[j] \\ \min\{C[j] + minCost(i-1, j-1), minCost(i, j-1)\} & \text{otherwise} \end{cases}$$

minCost(i,j) : min cost of an occurrence of X[1..i] as a subsequence of Y[1..j] or ∞ if no such occurrence

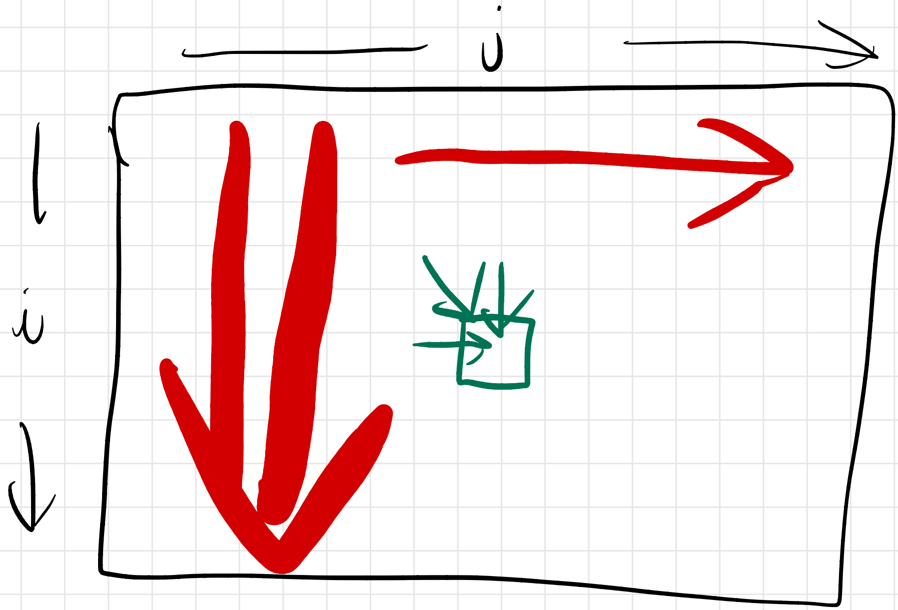
4349 Midterm ~~Spring~~ Fall 2017

Given $A[1..m]$ & $B[1..n]$.

Find length of the longest common subsequence. (LCS)

$LCS(i,j)$: length of LCS of $A[1..i]$ & $B[1..j]$.

$$LCS(i,j) = \begin{cases} 0 & \text{if } i=0 \text{ or } j=0 \\ \max\{LCS(i-1,j),\ LCS(i,j-1)\} & \text{if } A[i] \neq B[j] \\ \max\{LCS(i-1,j),\ LCS(i,j-1),\ 1+LCS(i-1,j-1)\} & \text{o.w.} \end{cases}$$



LCS[0... m, 0... n]

"row-major-order"

"by increasing i index,
by increasing j index"

top down left-to-right
"row-by-row, column by column