HW6 OH Sat. Midterm Next Sat 10/30

15-2 Longest palindrome subsequence of a given string.

$X = x_1 x_2 \cdots x_n$

$\bar{X} = x_n x_{n-1} \cdots x_1$

answer = LCS$(X, \bar{X})$

Requires explanation/proof ∵ DP step 1

Z = longest palindrome = OPTSOL(X)

$X =$ x_1 $\boxed{x_2}^{z}$ $\cdots$ $\boxed{x_k}^{z}$ $\cdots$ $\boxed{}^{z}$ x_{n-1} $\boxed{x_n}^{z}$

$\bar{X} = x_n$ x_n $\cdots$ x_k $\cdots$ x_2 x_1

goal: Z = palindrome subseq(X) $\iff$ Z commonsubseq$(X, \bar{X})$

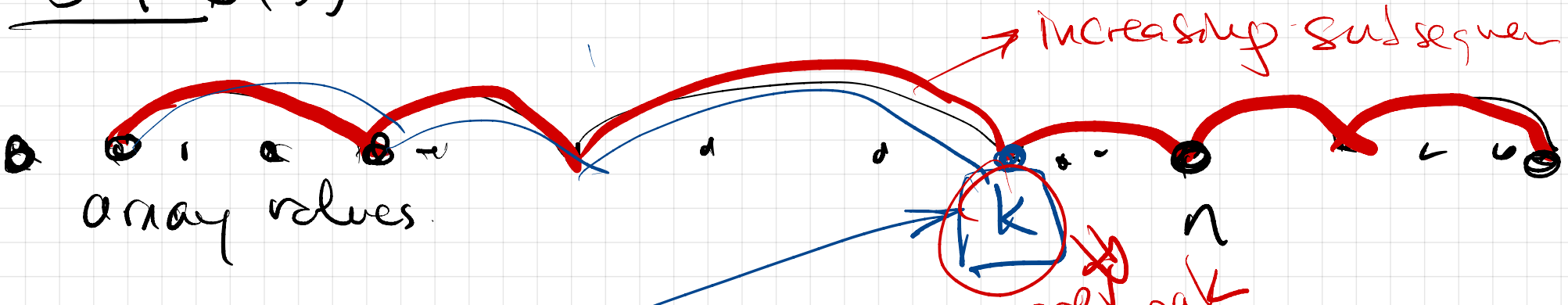
15-10 - class discussion Wed 10/27

- submit after that (Thu night/Fri morning)
- broker fee changing stock allocation F_2 (fixed fee)
- dont change any allocation $F_1 < F_2$

d) max $ in a stock $\leq$ VB = given

still DP?

15.4-6 (5)



C[n] = longest subseq ends in index n

② search for previous k index — linear search 15.4-5 ⇓ $\Theta(n^2)$ runtime

- C[k] = longest ending in k
- x[k] ≤ x[n] (xn can be added to C[k] seq)

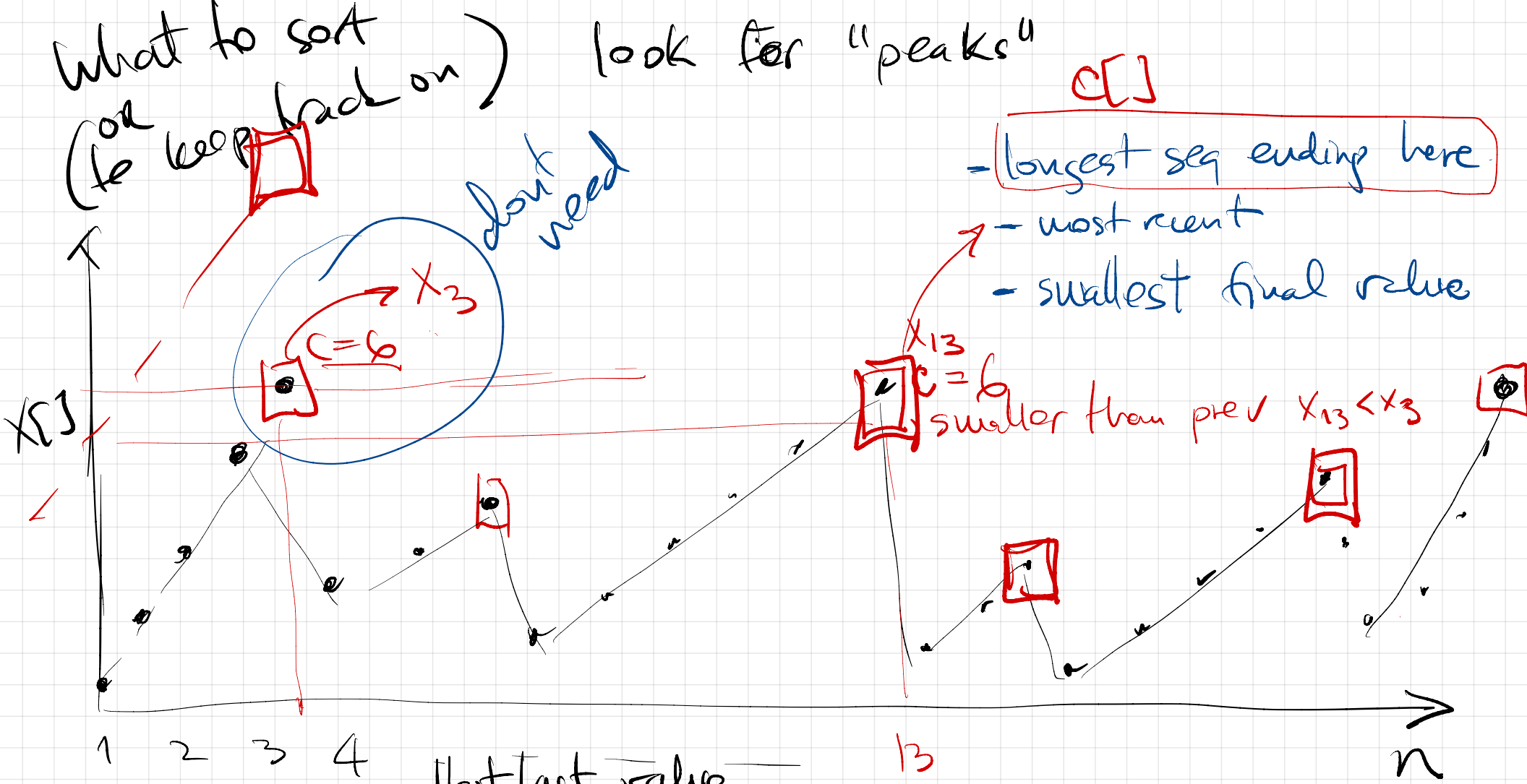
C[n] = C[k] + 1

logarithmic

15.4-6 ★ $\Theta(n \log n)$ RT. ⇒ search space must be (almost) sorted per use list of peaks sorted.

What to sort (or to keep track on)

look for "peaks"



smallest last value

B[i] = in subseq (c=i)

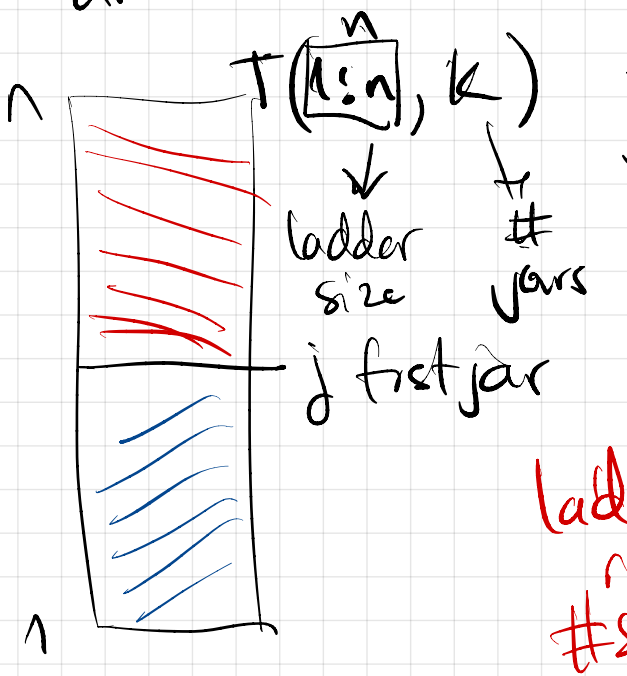
- remove peaks not useful (- same obj, larger last val
- smaller obj)

15.4-6

① peak list is sorted

② - peak we need (k) must be here — update peak list after k-select must O(log n)

Jar on Ladder PB. Search for step j to put first jar



jar breaks $\xrightarrow{\text{subPB}}$ $T([1:j-1], k-1)$ bottom

jar doesn't break $\xrightarrow{\text{subPB}}$ $T([j+1:n], k)$ top

$n-j$

ladder $[j+1:n]$ = ladder $[1:n-j]$

#steps on ladder = same size of ladder = n-j (range)

#of trials q

$T(n,k) =$ search for j MIN { MAX { worst case: breaks $T(j-1, k-1)$; drive adversary: doesn't break $T(n-j, k)$ } } $+1$

MIN worst case "best worst case"

MIN intuition: balance the two subproblems difficulty

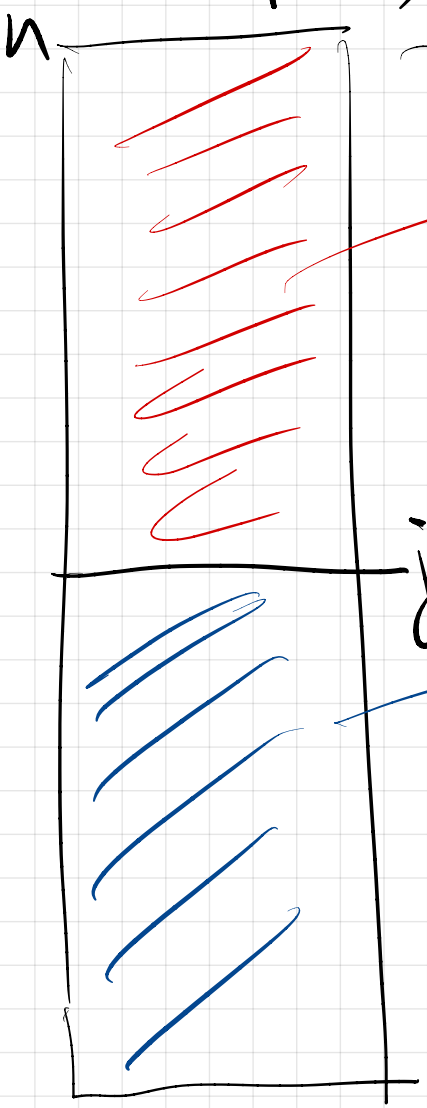
↳ MAX irrelevant

? $q \geq \log n$ OR IMPOSSIBLE

? $k \leq \log n$ OR binary-search

Dual recursion $R(k, q) = n$ = max ladder size

k = jars, q = trials



$n = ?$ j = last step of jar.

$\Rightarrow n-j$ = biggest ladder with ? jars ? trials

$R(?, ?)$

same as $R(j-1)$

$(j-1)$ = biggest ladder with ? jars ? trials

$R(?, ?)$

$n = R(k, q) = \underbrace{n-j}_{R(?,?)} + \underbrace{j-1}_{R(?,?)} + 1$

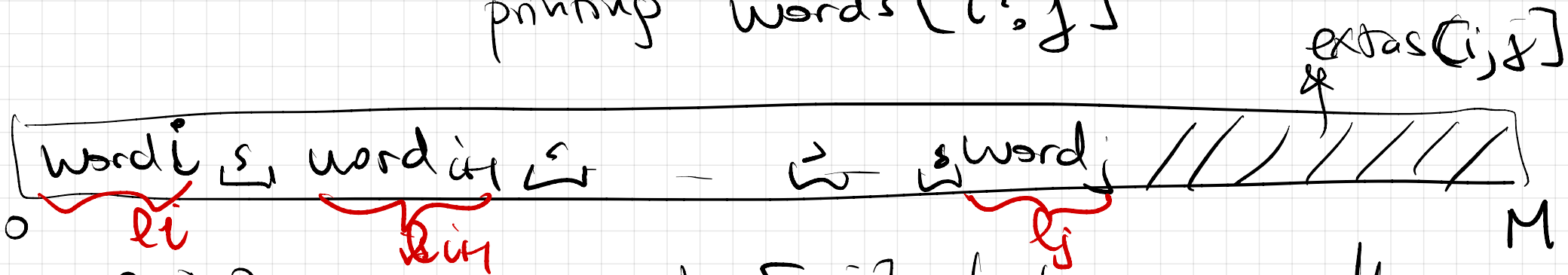
15-4 Printer neatly

Hint (precompute array / cache / or function)

access to values

?? $M - j + i - \sum_{k=i}^{j} \ell_k$??

extras[i,j] = extra spaces at end of a line printing words[i:j]



- DP assuming extra[i,j] look up penalty = $\sum$ extra[i,j]^3
- How to get extra[i,j] as 2) table or function call

OPTSOL = where to break for lines ≡ last word in each line.

line1 w_1 ... w_{t_1}

line2 w_{t_1+1} ... w_{t_2}

...

line g $w_{t_{g-1}+1}$ ... w_{t_g}

lastline w_{t_g+1} ... w_n — no penalty

C[??] = obj = sum_of_penalty_per_line =

search for the last break t_g = C[???] + penalty on prev line.

0 if w_n = last word