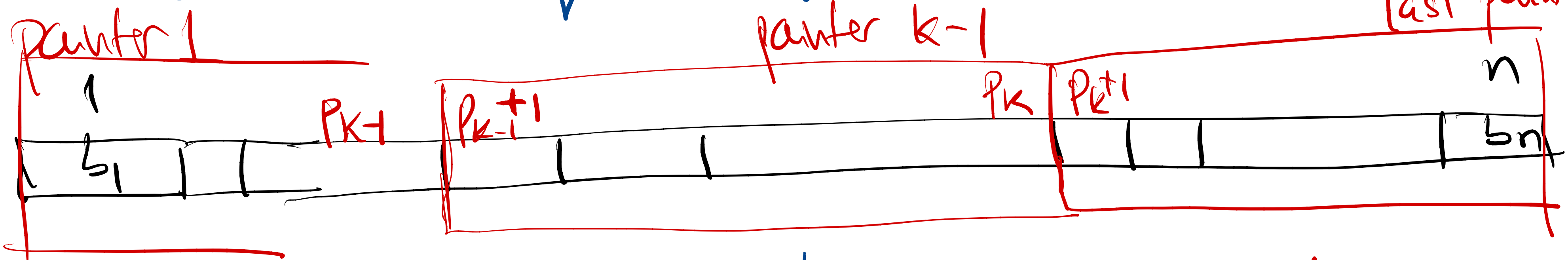
Painters Partition PB

n boards in seq of lengths $b_1, b_2 \dots b_n$



k painters, each paint contiguous boards W_{ij}

painter 1 paints $b_1 \dots b_{p_2}$
2 paints $b_{p_2+1} \dots b_{p_3}$
k paint $b_{p_k+1} \dots b_n$

parallel

time it takes for a painter $= \sum_{\text{painter}} \text{length(boards)}$ (hours)

Task: partition boards $\rightarrow$ painters
MIN time (longest painter)

$C[j, k]$ = min # hours it takes in parallel to paint boards

j → boards $1:j$

k → # painters

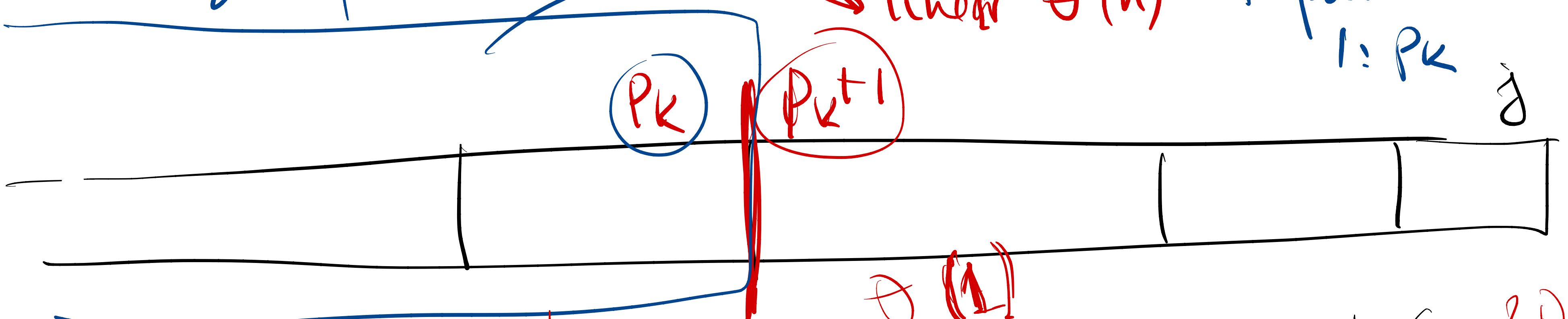
search for first board painter k paints p_k+1

$C[j,k]$ = search max p_k $\left\{ \sum_{t=p_k+1}^{j} b_t \;,\; C[p_k, k-1] \right\}$

$\sum_{t=p_k+1}^{j} b_t$ = # hours painter k takes → w.c. R.T $\Theta(j)$, best $\Theta(1)$ precom input

$C[p_k, k-1]$ = # hours (opt) to paint boards $1:p_k$ — linear · $\Theta(n)$

sub pb



$n \times k$ $C[\,]$ pbs $\times \log n$

hours last painter (precomp) 2D table

$$\Delta_{p_k} = \underbrace{C[p_k, k-1]}_{\text{prev } k-1 \text{ pain}} - \sum_{t=p_k+1}^{j} b_t$$

W_{p_k} precomputed

decreasing on when p_k → left

$\Theta(n \cdot \log_2(Bsum) \cdot k)$

Array B

$b_1 \quad b_2 \quad \boxed{b_3 \quad \dots} \quad \dots \quad b_n$

3 9

want $\Theta(1)$ access to $w(i,j) = \sum_{t=i}^{j} b_t$

$i \le j$

precomp $\Theta(n)$

space+KT

$\leftarrow F(j) = \sum_{t=1}^{j} b_t = b_1 + b_2 + \dots + b_j$ (partial sum)

on the fly $\Theta(1)$ $\leftarrow w(i,j) = F(j) - F(i-1)$

$$b_1 + b_2 + \dots + b_j - b_1 - b_2 - \dots - b_{i-1}$$

$$b_i + b_{i+1} + \dots b_j$$

$$f = \frac{\partial F}{\partial x}$$

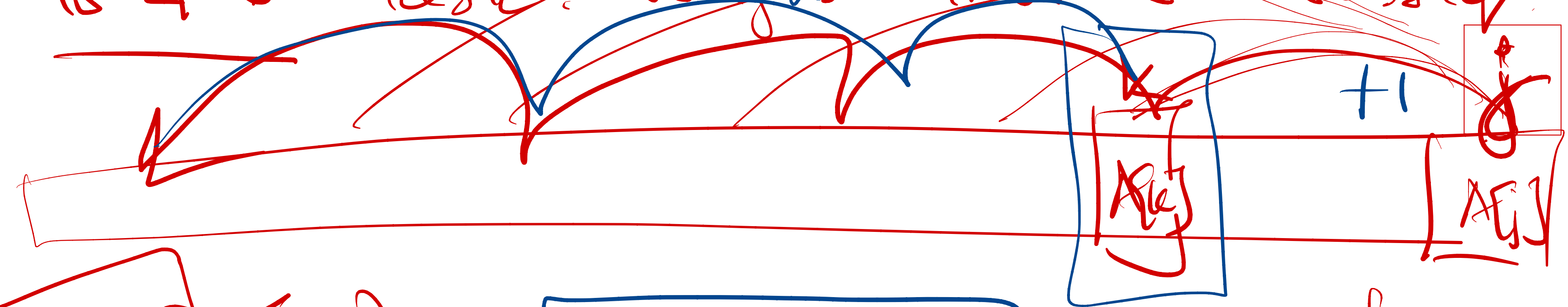
$$\int_a^b f(x)\,dx = F(b) - F(a)$$

$\Theta(n^2)$

$\Theta(n \log n)$ looking for $\Delta p_k = 0$?

DP - log(n) search

15.4-6 Task: longest increase subseq



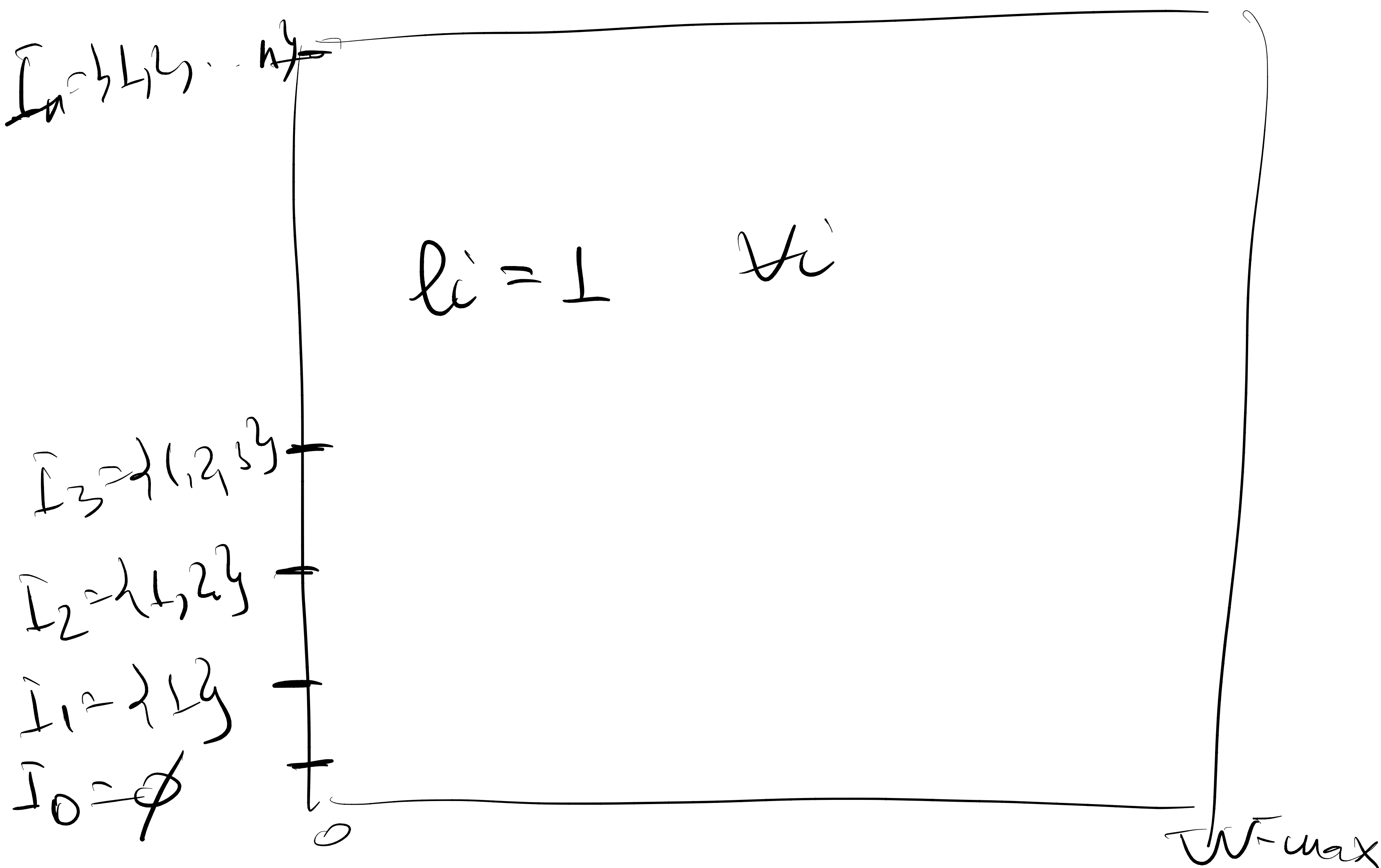
15.4-5 $C[j] = \boxed{\text{search for } k}$ → index of prev val in seq ending at j

- $\boxed{A[k] \le A[j]}$ constraint

seq $\boxed{C[k] \text{ max}}$ length

$$\boxed{C[j] = C[k] + 1}$$

→ one more element



W same

multiset => allows item repetitions

$$I_{full} = \{ \underbrace{1,1,1 \ldots}_{\ell_1}, \underbrace{2,2,2 \ldots 2}_{\ell_2}, \underbrace{3\,3\,3}_{\ell_3} \ldots \underbrace{n,n,n,n}_{\ell_n} \}$$

$$I = \{ \underbrace{1, \ldots 1}_{\ell_1}, \ldots \ldots, \underbrace{n, n, n}_{\ell_n - 1} \}$$

$C[W,$ multiset $\{1 - \ell_1$ times, $2\ \ell_2$ times, $3 \ldots\ j - \ell_j$ times$\}]$

pick last item j

$$= \begin{cases} C[W - w_j, \{1, \ell_1 \ldots\ j - (\ell_j - 1) \text{ times}\}] \\ \text{dont pick item } j \rightarrow \text{eliminate all } j \\ C[W, \{1 - \ell_1 \text{ times}, 2\ \ell_2 \text{ times} \ldots (j-1)\ \ell_{j-1} \text{ times}\}] \end{cases}$$

More DP problems

OBJ : { YES (1), NO (0) }

① word break : is <string> breakable into dict words?

dict = { I, like, sam, sung, samsung, mobile, ice, cream, icecream, man, go, mango }

<string> = "ilikesamsung" $A[1:n]$

k   n

i like sa $\boxed{m}$ sung

search for k $\left\{ \begin{array}{l} A[k+1:n] \in \text{dict} \\ C\{A[1:k]\} = \text{YES} \end{array} \right.$

- print <u>all</u> ways to break it
- OPTSOL = min # of words

② input integer

n = sum of integers (without order)

all possibilities = $P(n)$ PARTITION Function

$n = 5 = 4+1$

$= 3+2$

$= 3+1+1$

$= 2+2+1$

$= 2+1+1+1$

$= 1+1+1+1+1$

$6 = P(5)$

- compute $P(n)$
- list all sums

DP: $P(n) = \ldots P(<n)$

virgil idea

$P(n,k)$ = # ways to break n into integers $\le k$

- take k out $\Rightarrow 1 + P(n-k, k)$
- don't use k $\Rightarrow 0 + P(n, k-1)$

- all possibilities with exactly M integers in the sum

Parentheses $(A_1 A_2)(\ )(\ )\ (A_n)$

#ways $((())\ )\ )(((\ ())))$

$\boxed{\text{Catalan \#}}$