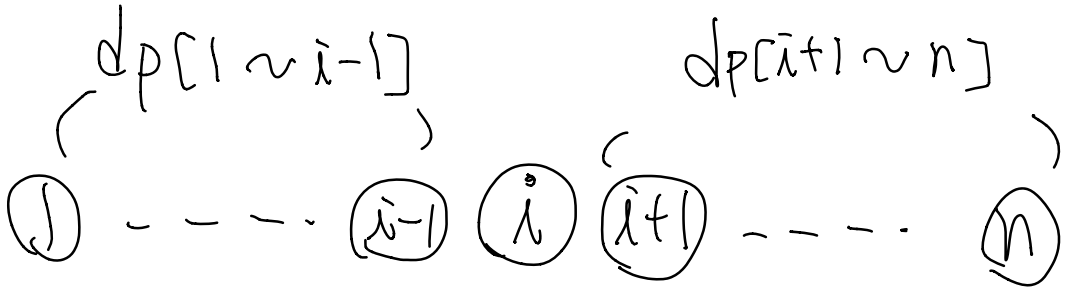
1 2 3 4 5 6 7 8 9 10

↑

$dp[1 \sim n] =$ $5 + \max(dp[1 \sim 4], dp[6 \sim 10])$

we choose ↓

$$dp[1 \sim n] = \min_{i \in [1 \sim n]} \left(i + \max(dp[1 \sim i-1], dp[i+1 \sim n]) \right)$$



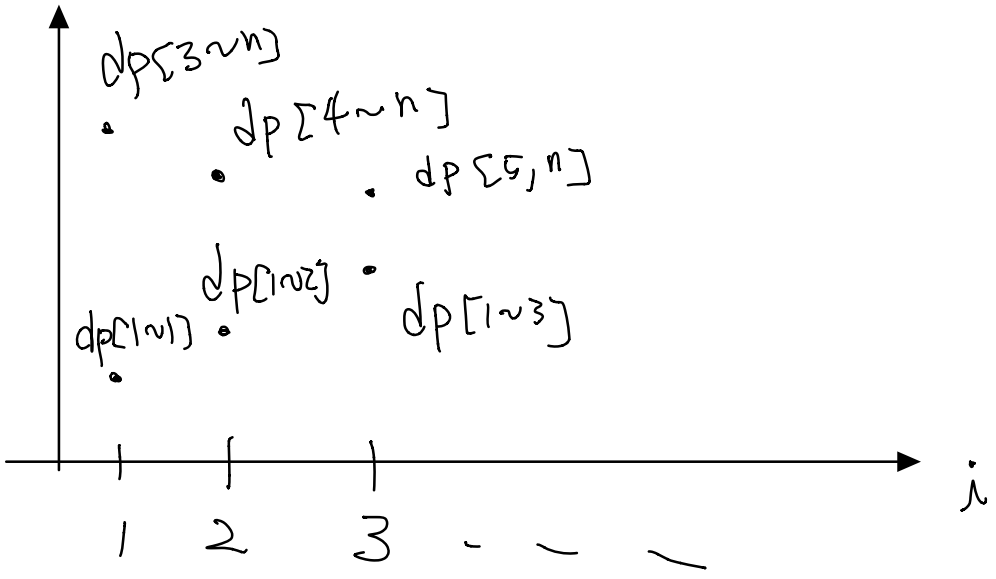
max(dp[1~1], dp[3~n])

∧ ∨

max(dp[1~2], dp[4~n])

∧ ∨

max(dp[1~3], dp[5~n])



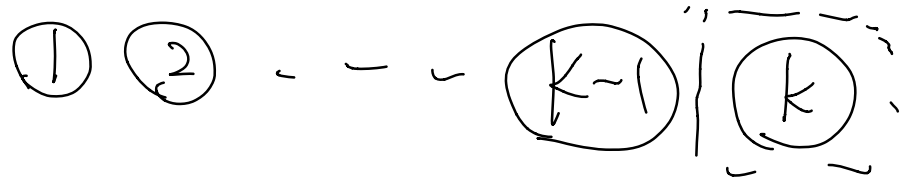
Proof: $dp[1 \sim 1], dp[1 \sim 2] \ldots$

$dp[1 \sim i]$ is an monotonic increasing ($\leq$) sequence.

(Proof by contradiction)

Assume

$$dp[1 \sim k-1] > dp[1 \sim k]$$

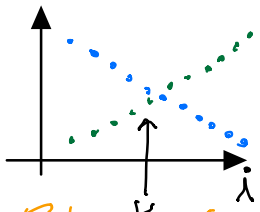


we can imagine adding k and use $dp[1 \sim k]$ strategy to play the game. So $dp[1 \sim k-1]$ becomes smaller, violate that it already is the minimum $\Rightarrow$ $dp[1 \sim k-1] \leq dp[1 \sim k]$

We already know we want to compute:

$$dp[a \sim b] = \min_{a \le i \le b} \left(i + \max(dp[a \sim i-1], dp[i+1 \sim b]) \right)$$

Observe: $dp[a \sim i-1]$ is a weakly increasing (as i increases) ($\le$) seq



$dp[i+1 \sim b]$ is a weakly decreasing ($\ge$) seq

Edge case: $b - a = 1$, $\min(a + dp[b \sim b], b + dp[a \sim a])$
$\Rightarrow$ we set $k = a$ is OK

We must be able to find a k s.t.

$$k = \max\{ i : dp[a \sim i-1] \le dp[i+1, b] \}$$

(i.e. $dp[a \sim k+1-1] > dp[k+1+1, b]$)

Case 1: $a \le i \le k$: $\max(dp[a \sim i-1], dp[i+1, b]) = dp[i+1, b]$ — (1)

Case 2: $k < i \le b$: $\max(dp[a \sim i-1], dp[i+1, b]) = dp[a \sim i-1]$ — (2)

By ①, ②,

$dp[a \sim b]$ can be divided into 2 parts

$\underline{a \le i \le k}$ and $\underline{k < i \le b}$

$$dp[a \sim b] = \min_{a \le i \le b} (\cdots)$$

$$= \min \left(\min_{a \le i \le k} (\cdots), \min_{k < i \le b} (\cdots) \right)$$

define dp_1 | define dp_2

using ①

$$dp_1 = \min_{a \le i \le k} \left(i + dp[i+1, b] \right)$$

→ Use deque to implement

using ②

$$dp_2 = \min_{k < i \le b} \left(i + dp[a \sim i-1] \right)$$

more elements ↓

$dp[a \sim i-1] \ge dp[a \sim i-1-1]$

∵ as i increases,

choose $i = k+1$ (must be the minimum)

$$= k+1 + dp[a \sim k+1-1]$$

$$= k+1 + dp[a \sim k] \rightarrow O(1) \text{ time}$$

The remaining part is:

how to compute ① k ② dp_1

in $O(1)$ time (or amortized $O(1)$)

For ①, observe

$$k = \max\{i : dp[a \sim i-1] \le dp[i+1, b]\}$$

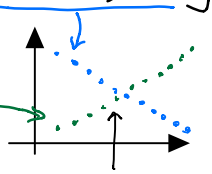


Illustration:

$dp[a \sim k-2]$ $dp[a \sim k-1]$ $dp[a \sim k]$

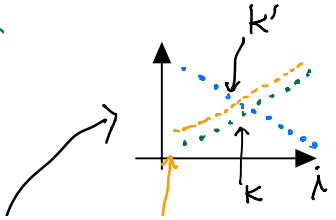
$\cdots \cdots$ $\wedge 1$ $\wedge 1$ $\vee$

$dp[k, b] \ge dp[k+1, b] \ge dp[k+2, b]$

If a becomes smaller (say $a' = a - 1$)

$$dp[a' \sim k-1] \ge dp[a \sim k-1]$$

$\because$ a' a $\cdots\cdots$ $k-1$



Let $k' = \max\{i : dp[a' \sim i-1] \leq dp[i+1 \sim b]\}$

Therefore, $k' \leq k$

because $dp[a' \sim i-1]$ becomes larger

$\Rightarrow$ k will ONLY move left (smaller)

as a becomes smaller. $\Rightarrow O(1)$ amortized, because k will only step on each position once

② $dp_1 = \min_{a \le i \le k} (i + dp[i+1 \sim b])$

Initial step: $a = b-1$, $dp_1 = a + dp[b \sim b]$

Next step: $a' = a-1$, we have to find a new k'

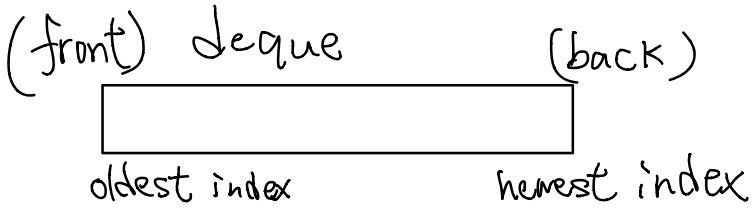
And if $k' < k$, we want to compute in $\min_{a' \le i \le k'} (i + dp[i+1 \sim b])$

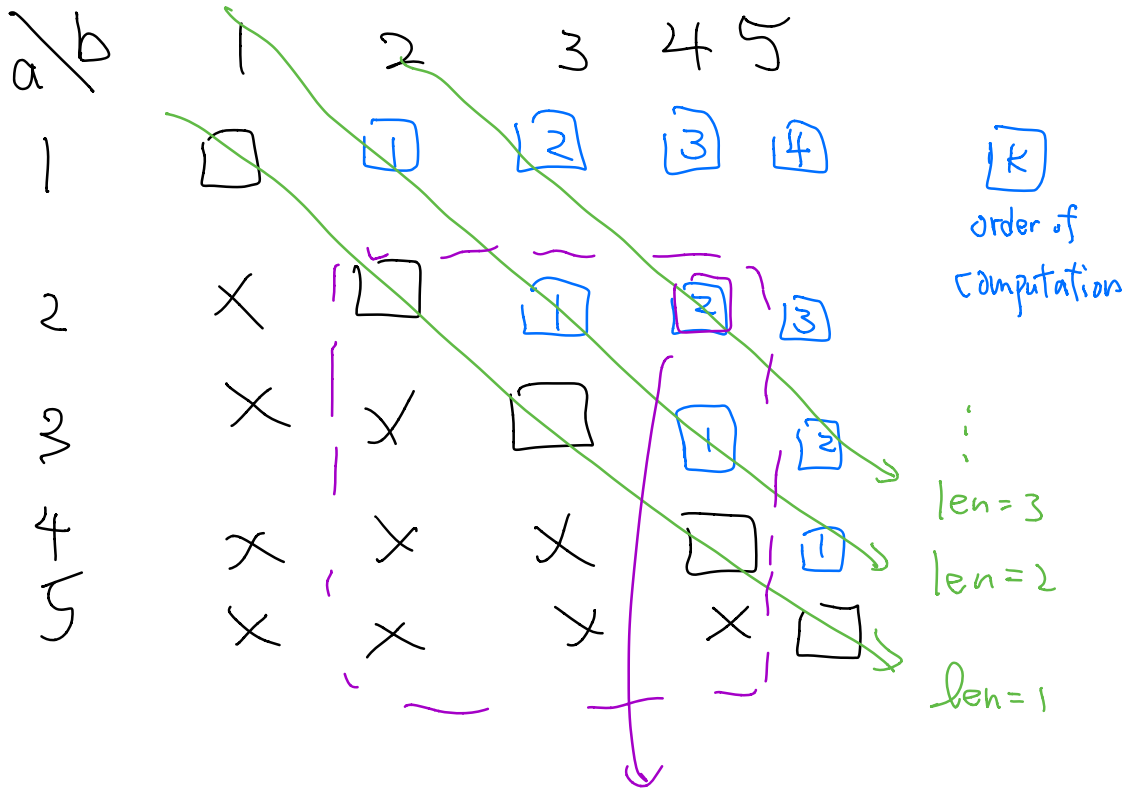
In $[a', k']$ window, we need to do 2 things.

① Throw away $i > k'$ (from front)

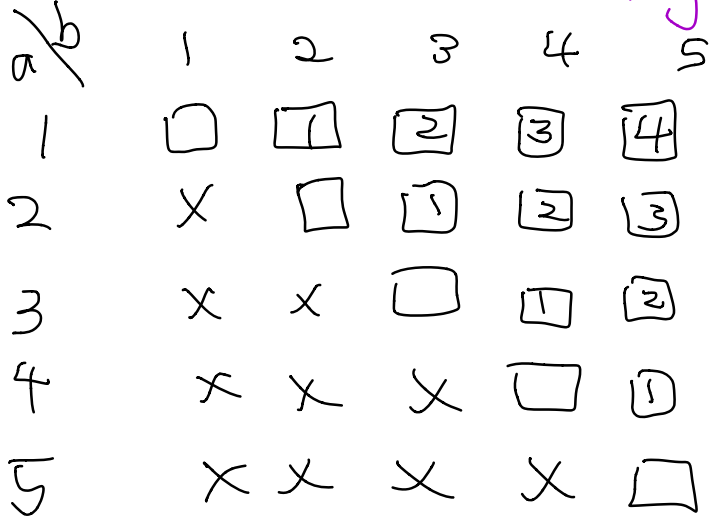
② put $(a', a' + dp[a'+1 \sim b])$ (to back)

in the back of an increasing deque





OR



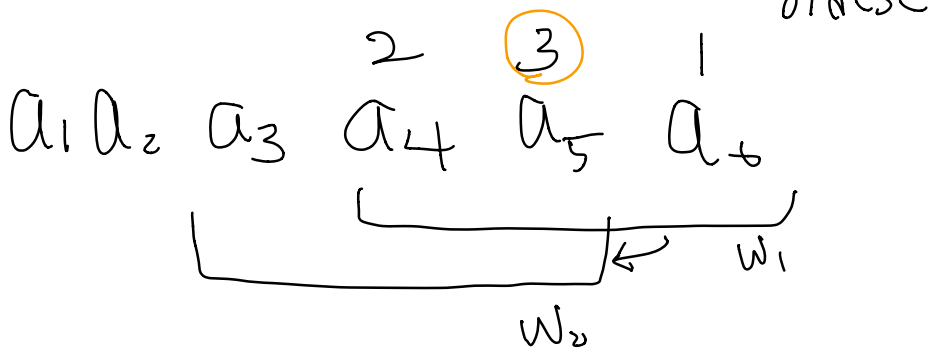
← row based should be faster

⇒ But this problem cannot be solved in row-based (See dp1 definition)

len = 1

Recap: Sliding window minimum

Ex:



In w_1: