Let's start at 9:02PM

L65
Dynamic Programming : Mixed Problems 1

Join Discord - https://bit.ly/ly-discord

RECAP

Let’s dive right into it

1. Make the Fence Great Again

N = 3

A = [2, 2, 3]

B = [4, 1, 5]

⇒ [3, 2, 3]

⇓

4

⇒ [2, 4, 3] ✓

1 + 1 = 2

N = 3

A = [2, 2, 2]

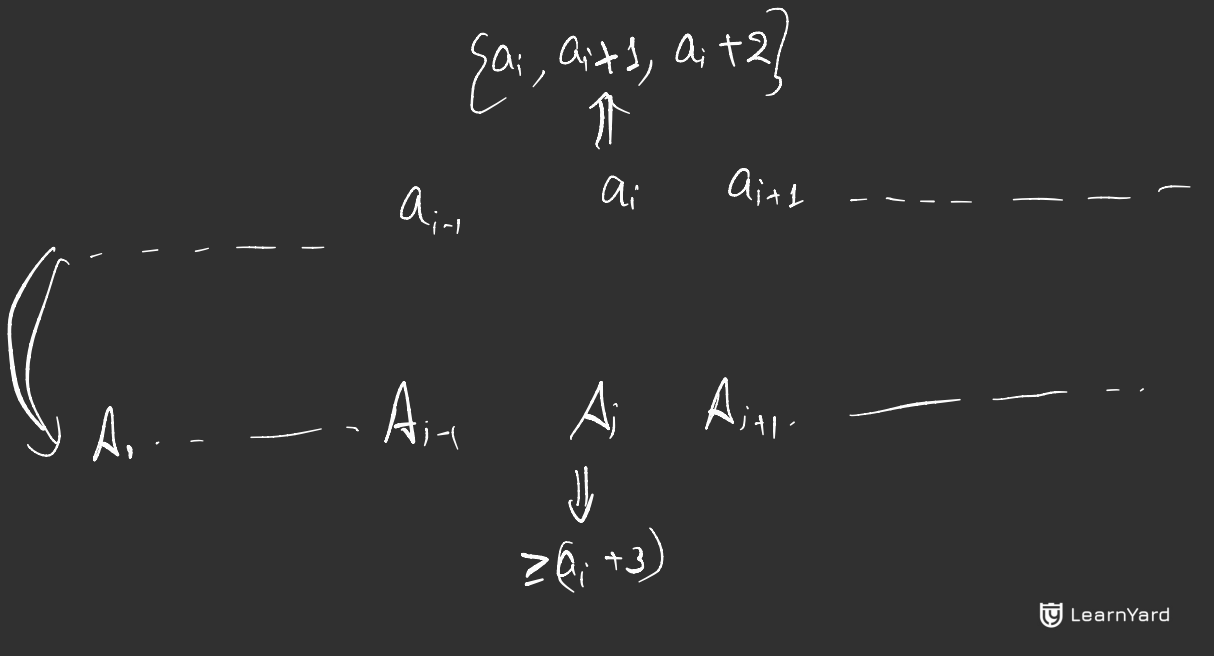
B = [3, 10, 6]

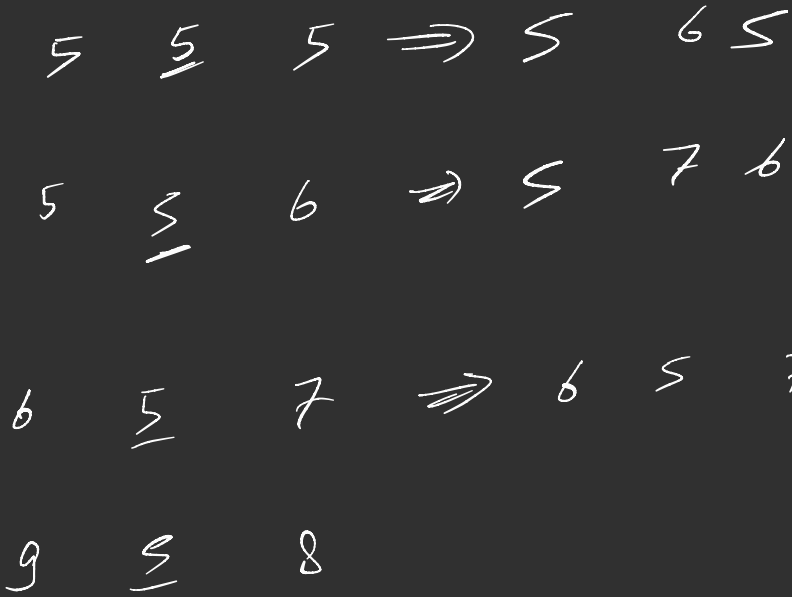
[3, 2, 3]

3 6

ans = 9

Intuition





$a[i] \Longrightarrow 0$ cost

$a[i] + 1 \Rightarrow 1 * b[i]$ cost

$a[i] + 2 \Rightarrow 2 * b[i]$ cost

$dp[i]$ = ans for $a[i \text{----} n-1]$

$dp[i][j] \Rightarrow$ ans for $a[i \text{----} n-1]$

0/1/2 given $a[i]$ is going to be incremented exactly j times.

$f(i, j)$ curCost = $j * b[i]$

// $a[i] + j$ (0/1/2)

future Cost ~ $f(i+1, 0) \Rightarrow a[i+1]$

↪ $f(i+1, 1) \Rightarrow a[i+1] + 1$

↪ $f(i+1, 2) \Rightarrow a[i+1] + 2$

Solution

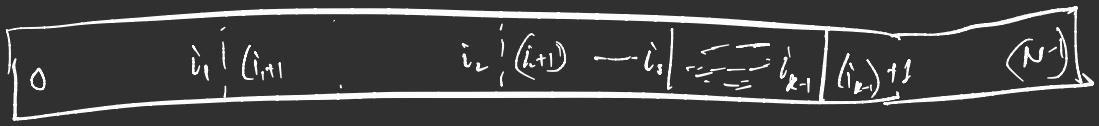
Let’s implement

2. Beautiful Partitioning

0-based

$i_1, i_2, i_3 \; - - \; - \; i_{k-1}$

$i_1 < i_2 < i_3 \; - - - - \; i_{k-1} < N-1$



$pre[i_1] > 0$

$pre[i_2] > pre[i_1]$

$pre[i_3] > pre[i_2]$

$pre[i_{k-1}] > pre[i_{k-2}]$

$pre[N-1] > pre[i_{k-1}]$

$0 < [pre[i_1] < pre[i_2] \text{ ---- } pre[i_{k-2}] < pre[i_{k-1}]] < pre[N-1]$

1) All pre[i] values are positive.

2) All pre[i] values should be less than pre[N-1]

3)* They should form an increasing subsequence (K-1)

Intuition

pre[0], pre[1], . . . pre[N-2], pre[N-1]

```
vector<int> V;
for(int i = 0; i < n; ++i)
    if (pre[i] > 0 && pre[i] < pre[N-1])
        V.push_back(pre[i]);

if (LIS(V) >= K-1)
    YES
else NO;
```

Solution

Let’s implement

Thank You!

Reminder: Going to the gym & observing the trainer work out can help you know the right technique, but you'll muscle up only if you lift some weights yourself.

So, PRACTICE, PRACTICE, PRACTICE!