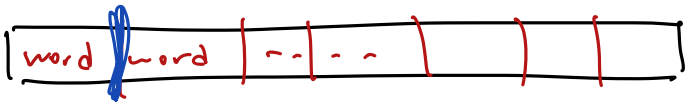
Monday is Labor Day

Dynamic Programming

Text segmentation:



Where does the first word end?

Splittable(i) = Is A[i..n] splittable into words?

$O(n^2)$ time

Longest Increasing Subsequence

3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6 2 7

Is this the next element of the LIS, given that previous element was here

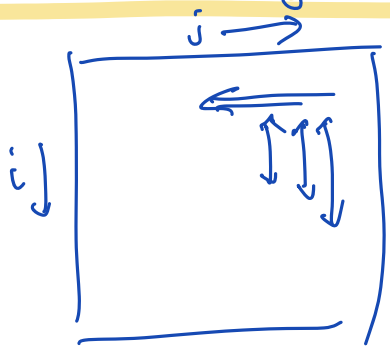
What do we need for DP full credit?

1. English specification
2. Recurrence
3. Iterative details

—or—

4. Pseudo code + time

LIS(i,j) = length of longest incr subseq of A[j..n] all bigger than A[i]



$O(n^2)$

Longest Increasing Subsequence. Take 2

$-\infty$ 3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4 6 2 7

A[i]

What is the ~~first~~ next element of LIS?

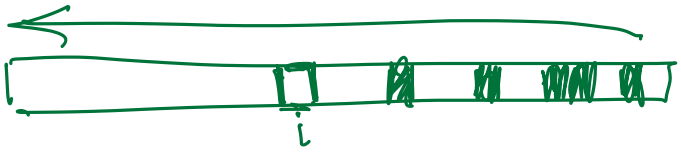
What is second element in LIS whose first element is here?

A[0] ← $-\infty$
return LIS(0) − 1

$LIS'(i)$ = length of longest increasing subseq of $A[i..n]$ starting with $A[i]$

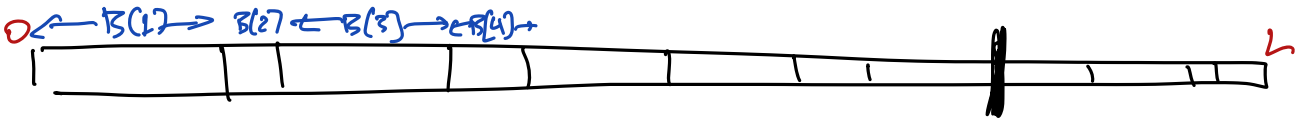
$$LIS'(i) = 1 + \max\{LIS'(j) \mid i<j\le n \text{ and } A[j]>A[i]\}$$

$[\max \emptyset = 0]$



$O(n^2)$ time

Woodcutter's Problem



Long plank with cut marks

Sawmill will make <u>one</u> cut, cost = length of plank

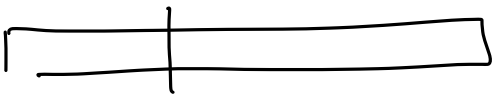
What is the cheapest way to make <u>all</u> cuts?

Given array $B[1..n]$ of board lengths

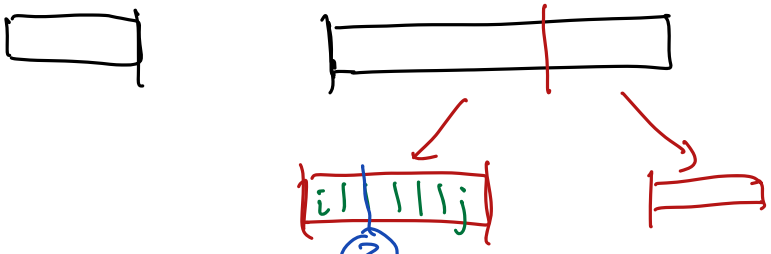
↓

$X[1..n]$ of cut positions $\quad X[j] = \sum_{i \le j} B[i]$

$B[i] = X[i] - X[i-1]$

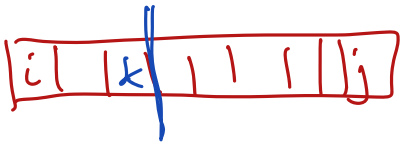


$Length(i,j) = X[j] - X[i-1]$

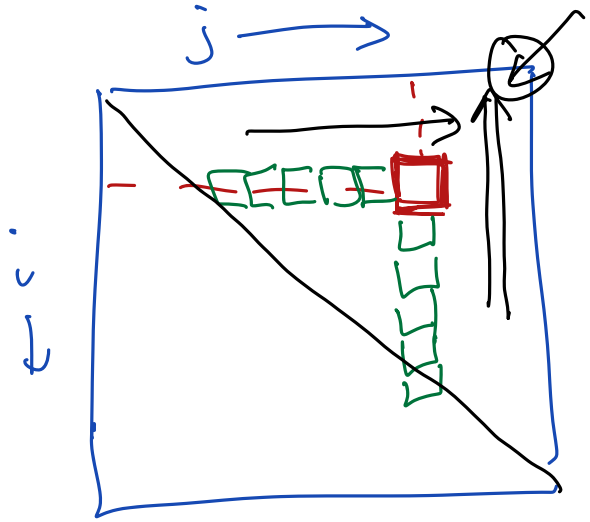


$WoodCut(i,j)$ = min. cost to cut plank containing boards $i..j$ into individual boards.

$$WoodCut(i,j) = \begin{cases} \min\{Length(i,j) + WoodCut(i,k) + WoodCut(k+1,j) \mid i \le k < j\} \\ 0 \quad \text{if } i = j \end{cases}$$



k = index of last board in left chunk



$O(n^3)$