Admin: Hw 1 – due 9pm

Hw 2 – out today, due next Tue

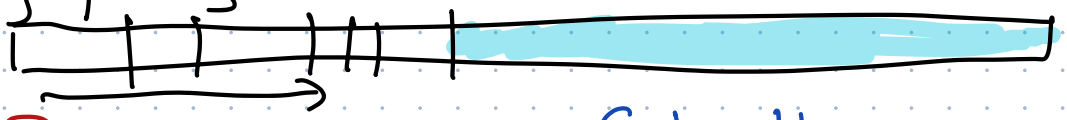
Sources and collaborators

Dynamic Programming

What are the subproblems?

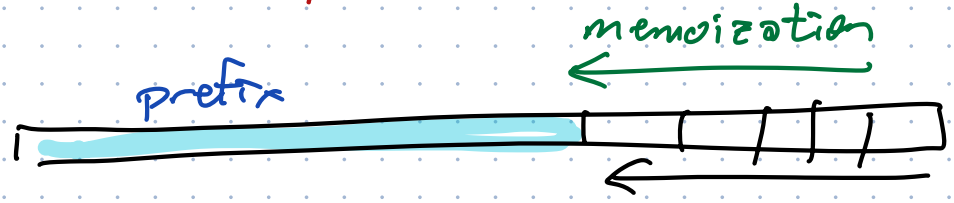
inputs to

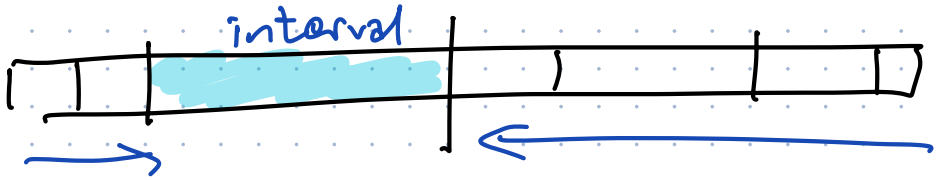
String splitting



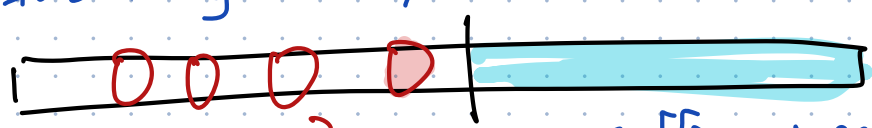
Decisions: left to right word boundaries

Subproblem: suffix





Longest Increasing Subsequence:



decisions

suffix + one element

Longest Common Increasing Subsequence

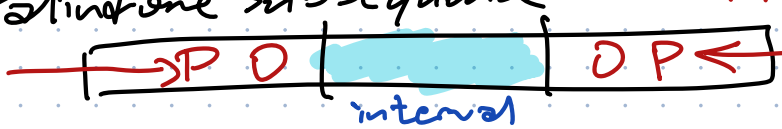
A ALEXANDERHAMILTON

B ALMONDALLERGY

two suffixes + Fencepost

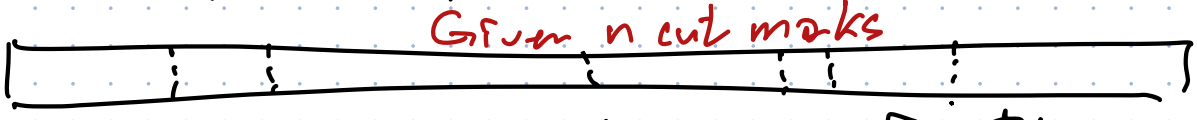
Longest Palindrome Subsequence

MADAMIMADAM



$P \to \varepsilon \mid a \mid aPa$

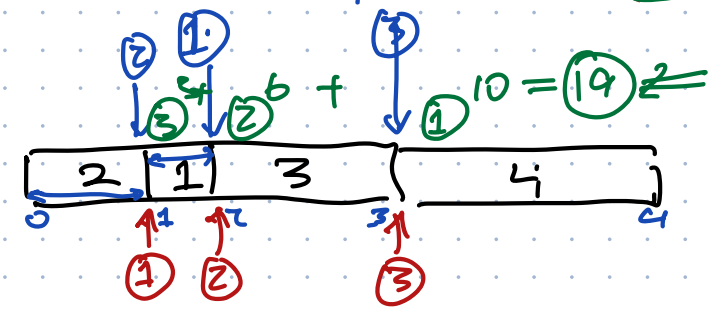
Woodcutter's problem



Sawmill can cut any L-foot board once for $L

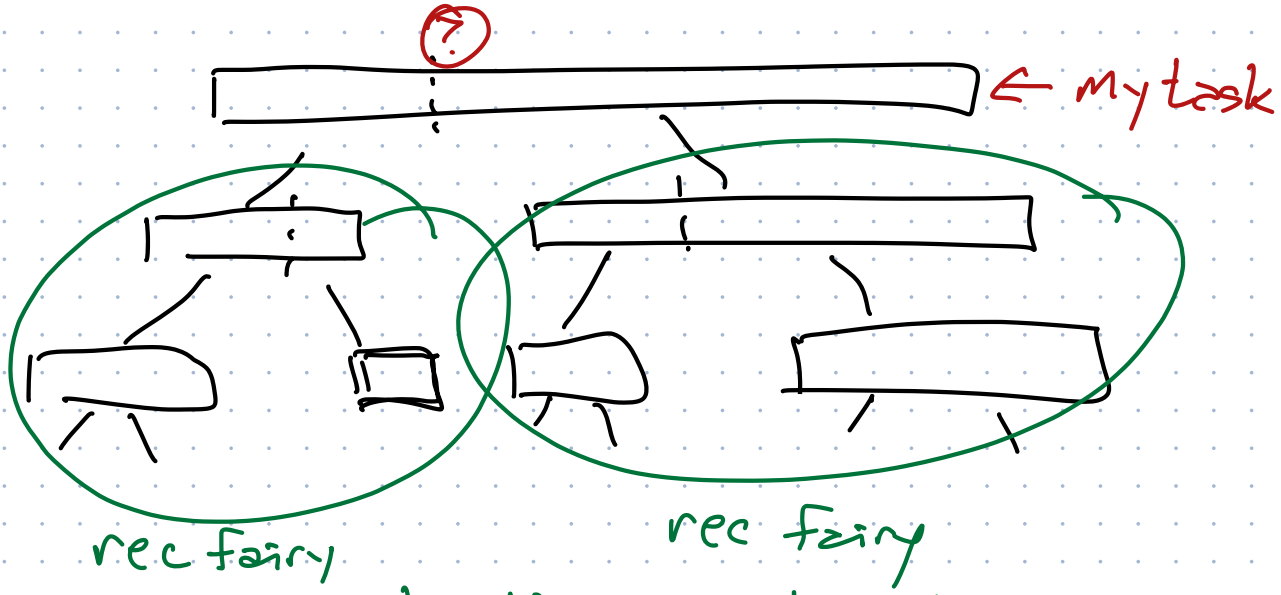
Find order of cuts with min total cost

3+10 + 7 = 20

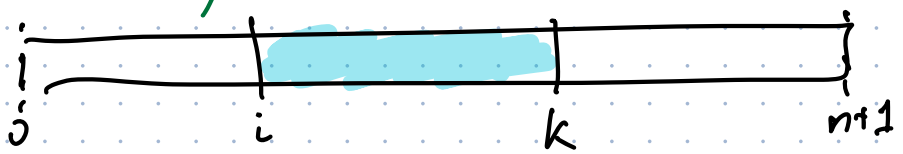


L=[2,1,3,4]

10+8 + 7 = 25



subproblems = intervals

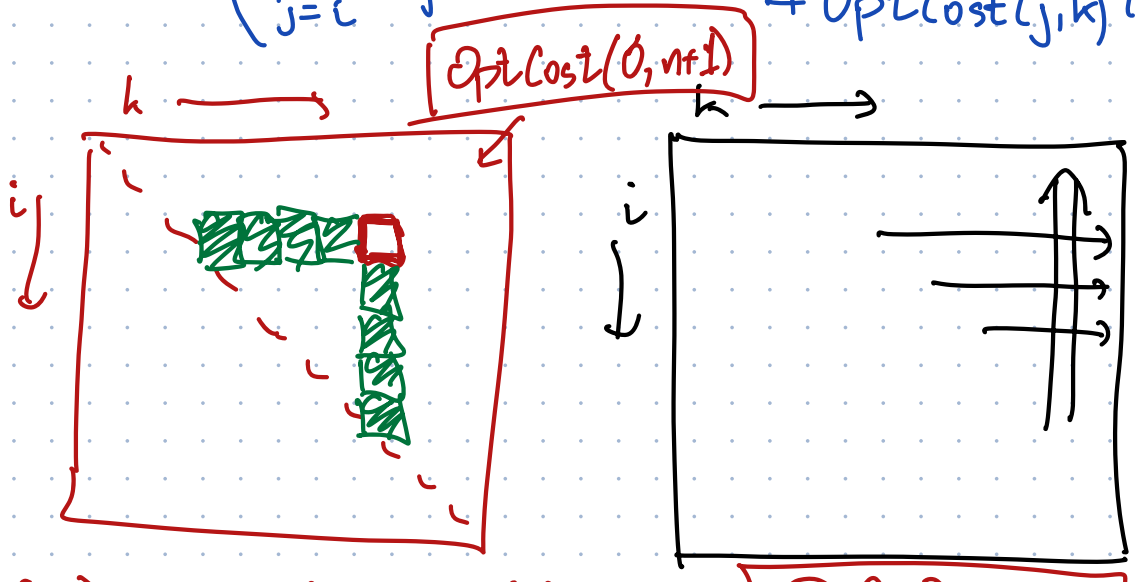


OptCost(i,k) = min. cost to cut up segment from cut i to cut k.

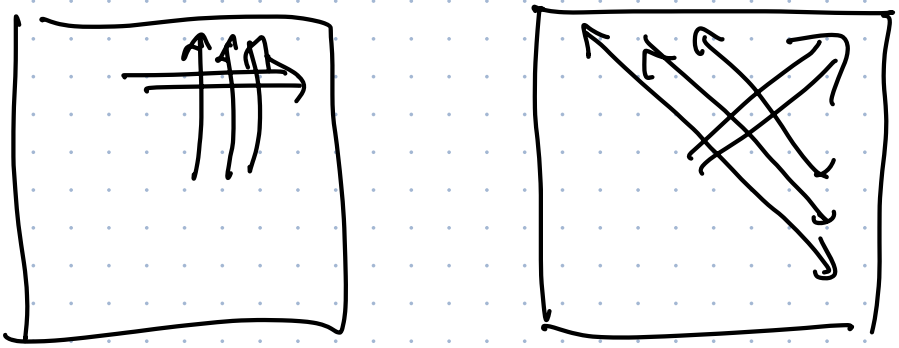
Input: L[0..n] = array of board lengths

L[i] = length between cut i and cut i+1

$$OptCost(i,k) = \begin{cases} 0 & \text{if } k = i+1 \\ \sum_{j=i}^{k-1} L[j] + \min\{ OptCost(i,j) + OptCost(j,k) \mid i<j<k \} \end{cases}$$



$O(n^2)$ subprobs $\times$ $O(n)$ time each $= O(n^3)$ time



We can do better! $O(n^2)$ time Knuth-Yao

$O(n \log n)$