Welcome to CS473 (CS 498 DL1)

Algorithms

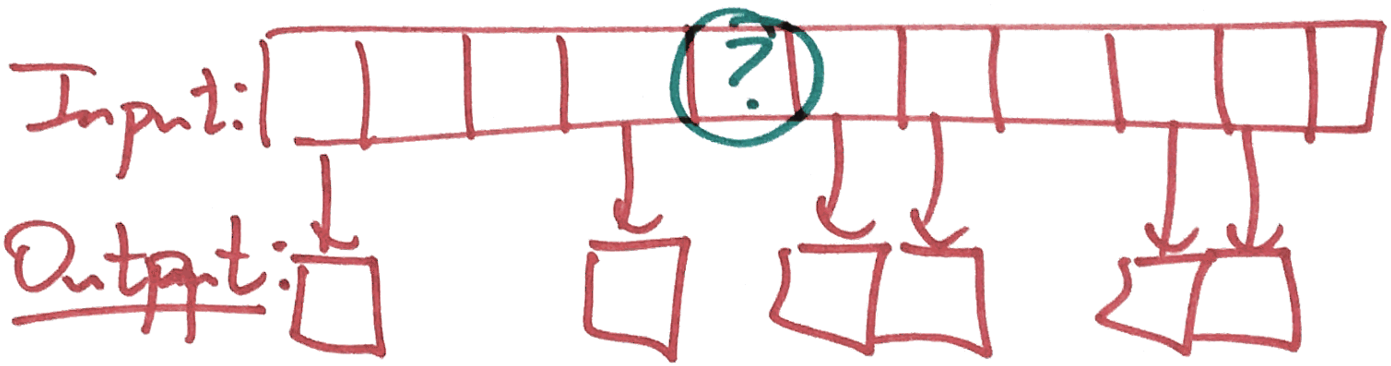
Jeff

Dynamic Programming

Optimization/Construction problem

→ Series of decisions

↑ what decisions?



Formulate decisions as a recursive problem

→ specify

→ recurrence — CORRECT

Make the recurrence iterative — EFFICIENT

Longest Increasing Subsequence

A = 1 8 7 8 7 8 7 8 4 1 1 4 8 9 4 9 11 8 9 8 9 8 Hey!

i j

1) Seq. of decisions

Is this in the LIS assuming prev decisions are correct?

2) **Problem:** LIS(i, j) = Length of Longest increasing subsequence of A[j..n] such that all entries are larger than A[i]

LIS(A) = LIS(0, 1)

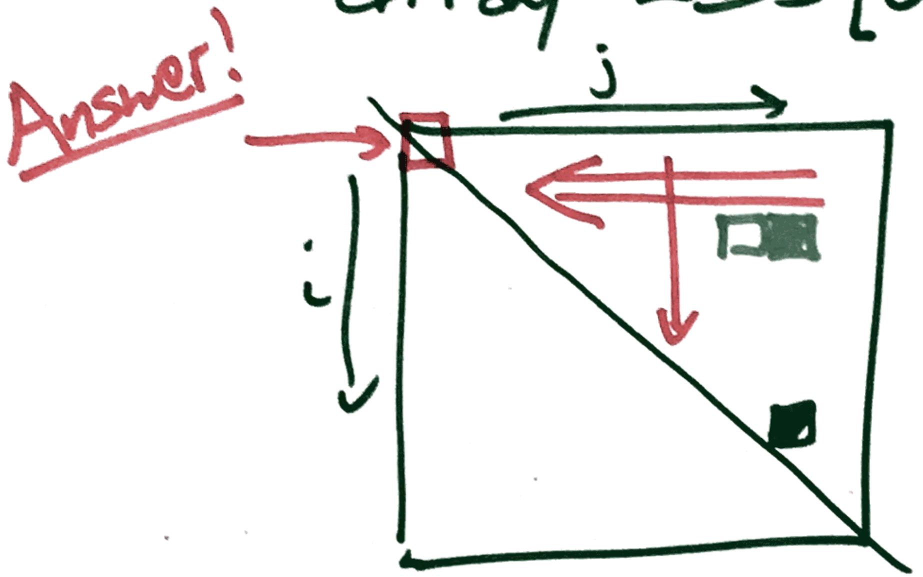
③ Recurrence:

$$
LIS(i,j) = \begin{cases} 0 & \text{if } j > n \\ LIS(i,j+1) & \text{if } A[j] \le A[i] \\ \max \begin{Bmatrix} LIS(i,j+1) \\ 1 + LIS(j,j+1) \end{Bmatrix} & \text{otherwise} \end{cases}
$$

④ Iterative

Memoization structure:

array LIS[0..n, 1..n+1]



Order:

for $j \leftarrow n$ down to 1
  for $i \leftarrow 0$ to j-1

$\boxed{\text{Recurrence}}$

$\boxed{O(n^2) \text{ time}}$

Decisions: What's next in the output subseq?

Problem: $\boxed{LIS(i) = \text{Length of longest increasing subseq. of } A[1..n] \text{ that starts with } A[i]}$

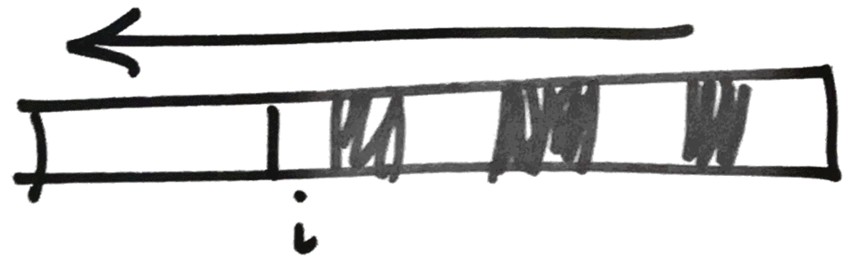
Global: $A[0] \leftarrow -\infty$

return $LIS(0) - 1$

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

where $\max \emptyset = 0$

LIS[1..n]



for $i \leftarrow n$ to 0

$\boxed{\text{recurrence}}$

$O(n^2)$ time