Problem Statement

Example:

\[
\begin{array}{cccccccc}
30 & 10 & 20 & 40 & 60 & 50 & 70 & 90 & 80 \\
\end{array}
\]
First Algorithm

- $A[0] = -\infty$ (for simplicity)
- $\text{LISBigger}(i, j)$: the LIS of $A[j \ldots n]$ w > $A[i]$
- What is top-level call?

$LIS(0, 1) - 1$

Think: $A[i]$ must be added with recent $A[j]$

if $j > n$ return $0$

skip = $\text{LISBigger}(i, j+1)$

if $A[i] \geq A[j]$ return skip

use = $\text{LISBigger}(j', j+1)$

return $\max$(skip, use)
Iterative Version

Decl LISB[0...n, 1...n+1]

▷ skip = LISB(i, j + 1)
▷ use = LISB(j, j + 1) + 1

for j = n ... 1 // (decreasing)
  for i = 0 ... j-1
    skip = LISB[i, j+1]
    else
      use = LISB[j, j+1] + 1
      LISB[i, j] = max(use, skip)

A[0] = -\infty

for i = 0 ... n
  LISB[i, n+1] = 0
Improved Algorithm

\[ \text{L1SFirst}(i) : \text{LIS w/ } i \text{ as 1st} \]

\[
\max \{ 1 + \text{L1SFirst}(j) : j > i \text{ and } A[j] > A[i] \}
\]

evaluate this

for all where this is true.

Good review: do iterative 😊
Iterative Version

\[ \text{LISFirst} = 1 + \max\{\text{LISFirst}(j) : j > i \ \text{and} \ A[j] > A[i]\} \]
Demonstration