

DP Algorithm for the Longest Increasing Subsequence

Input: array $X = [x_1, x_2, \dots, x_n]$

Output: length of longest increasing subsequence

- ▶ $Q[1] := 1$
- ▶ For $i = 2$ up to n :
 - ▶ IF $Pred[i] = \emptyset$ THEN $Q[i] := 1$
 - ▶ ELSE $Q[i] := \max\{Q[j] + 1 \mid j \in Pred[i]\}$
- ▶ Return $\max\{Q[1], Q[2], Q[3], \dots, Q[n]\}$

Note: to find the actual longest increasing subsequence, you have to do backtracing.