

>> Create an array `dp` of length `n`, where `n` is the length of the sequence.

>> Initialize all elements of `dp` to 1.

>> Iterate over all pairs of indices (i, j) such that $j < i$.

>> For each such pair, if the element at index `i` is greater than the element at index `j`, update `dp[i]` to be the maximum of its current value and `dp[j] + 1`.

>> After iterating over all pairs of indices, the maximum value in the `dp` array is the length of the longest increasing subsequence.

>> To reconstruct the subsequence itself, start at the index `i` with the maximum value in the `dp` array and iterate backwards through the array. At each step, if `dp[i]` is equal to `dp[j] + 1` and the element at index `j` is less than the element at index `i`, add the element at index `j` to the subsequence and update `i` to be `j`.

>> Reverse the subsequence to obtain the longest increasing subsequence in the original sequence.