

1. Consider the following “Longest Increasing Sublist” problem.

Input: A list of integers $L = [a_1, a_2, \dots, a_n]$.

Output: A sublist $L' = [a_{i_1}, a_{i_2}, \dots, a_{i_k}]$ such that $1 \leq i_1 < i_2 < \dots < i_k \leq n$ and $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ and k is maximum.

For example, if $L = [4, 1, 7, 3, 10, 2, 5, 9]$, then $L_1 = [1, 3, 5, 9]$ and $L_2 = [1, 2, 5, 9]$ are two optimal solutions, but $[1, 2, 3, 4]$ is not a solution (it takes integers from L out of order), $[1, 7, 3, 10]$ is not a solution (it is not increasing), and $[4, 7, 10]$ is not an optimal solution (it is not as long as possible).

Give a dynamic programming algorithm to solve the Longest Increasing Sublist problem. (HINTS: Any optimal solution for input $[a_1, a_2, \dots, a_n]$ either contains a_n , or it does not. Consider sub-problems whose solutions have last element a_k , for various values of k .)