question: Recall the dynamic program for longest increasing subsequence (LIS) for an input sequence of n numbers $a_1, a_2, \ldots, a_n$. If $L(i)$ is the length of the LIS that ends in and includes a_i, then $L(i) = 1 + \max\{L(j) \ : \ j < i \text{ and } a_j < a_i\}$

1. Give pseudocode that turns this formula for $L(i)$ into an algorithm for finding the *length* of the LIS of the original sequence. Use the ideas of dynamic programming.
2. What is the running time of your algorithm in terms of n?
3. Prove that the formula $L(i) = 1 + \max\{L(j) \ : \ j < i \text{ and } a_j < a_i\}$ correctly computes the LIS that ends in and includes a_i.
4. What is the longest increasing subsequence of the following input sequence?

$$0 \quad 8 \quad 4 \quad 12 \quad 2 \quad 10 \quad 6 \quad 14 \quad 1 \quad 9 \quad 5 \quad 13 \quad 3 \quad 11 \quad 7 \quad 15$$

questions:

1. Modify the dynamic program for the knapsack problem to find a set of items of maximum value whose total weight is *exactly* the capacity of the knapsack. Note that for a given set of items, there may not be a subset of items whose total weight is *exactly* the capacity of the knapsack; in this case, your algorithm should correctly say there is no solution.

   You should:

   (a) Define the dynamic programming table.
   (b) Give a recursive formula for an entry in the dynamic programming table.
   (c) Describe in words how to fill the dynamic programming table.
   (d) Give pseudocode for the final algorithm *including* how to find and return the items in the knapsack.

2. Give an $O(nt)$ dynamic programming algorithm for the following task:

   Input: A list of n positive integers $a_1, a_2, \ldots, a_n$ and a positive integer K. *Question:* Does some subset of the a_i's add up to K? (You can use each a_i at most once.)

   You should:

   (a) Define the dynamic programming table.
   (b) Give a recursive formula for an entry in the dynamic programming table.
   (c) Describe in words how to fill the dynamic programming table.
   (d) Give pseudocode for the final algorithm.
   (e) Give the running time of your algorithm.