EECS 336 Fall 2015
Homework Problem 5.1

Follow the four-step procedure in dynamic programming.
Step One: OPT(j) = "maximum number of feasible boxes that are contained by box j."

Step Two: Recurrence. For $j > i$

$$OPT(j) = \left\{ \begin{array}{l} OPT(i) + 1\,(h_i < h_j, w_i < w_j, l_i < l_j)\,, \\ OPT(i)(otherwise) \end{array} \right\}$$

Step Three: Base case $OPT(1) = 1$

Step Four: Iterative DP

Algorithm 1 Iterative DP to find maximum number of boxes

procedure MaxBoxes(h, w, l)
  Sort n boxes in increasing h, $h_i < h_{i+1}$
  Initialize memo[i] $= 0$ for all $i \leq n$
  Set memo[1] $= 1$
  for $i = 1$ up to n **do**

$$memo(i+1) = \left\{ \begin{array}{l} memo(i) + 1\,(h_i < h_{i+1}, w_i < w_{i+1}, l_i < l_{i+1})\,, \\ memo(i)(otherwise) \end{array} \right\}$$

  end for
end procedure

Correctness:

By induction on recurrence. Suppose the algorithm gets correct answer up to box i. According to the recurrence, number of boxes contained by $i+1$ will be one more than at step i (when all of h, w, l fit the criterion, the increment counts for box $i+1$ itself), or will be the same as at step i (when at least one of h, w, l does not fit the criterion, $i+1$ cannot contain i. Thus, the recurrence is correct from box i to $i+1$.

Runtime Analysis:

Sorting n boxes requires $O(n \log(n))$. During each of the n recurrence steps, constant time is used to compare and update memo table. Thus, total time in recurrence is $O(n)$. Thus, total runtime is $O(n \log(n))$.