15.4-6 : Let $L[i]$ denote the length of the longest monotonically increasing sequence (LMIS) taken from the sequence of the first $i$ elements $a_1, a_2, \ldots, a_i$ of the $n$ element list. Further, let $A[i]$ denote the smallest element that can end a LMIS on the first $i$ items. We show how to compute $L[n]$ to obtain the LMIS in $O(n \log n)$ time by incrementally computing $L[i + 1]$ and $A[i + 1]$ from $L[i]$ and $A[i]$. We leave it as an exercise to extend this to obtain an actual representative sequence of length $L[n]$.

Suppose we know $L[i]$ and $A[i]$ for some $i$. To compute the values for $i + 1$, consider two cases with respect to element $a_{i+1}$.

**Case 1:** $a_{i+1} \geq A[i]$. In this case we can append $a_{i+1}$ to the optimal list for $i$ elements to get:

$$L[i + 1] = L[i] + 1$$
$$A[i + 1] = a_{i+1}$$

The update in this case takes $O(1)$ time.

**Case 2:** $a_{i+1} < A[i]$. If $L[i + 1] > L[i]$, then the corresponding list that achieves $L[i + 1]$ would have a length $L[i + 1] - 1$ prefix constructed from the first $i$ items that ends with an element less than or equal to $a_{i+1}$, and thus less than $A[i]$, which is a contradiction. Thus,

$$L[i + 1] = L[i]$$

For $A[i + 1]$, we have that $A[i + 1] = a_{i+1}$ if and only if for some $j < i$, $L[j] = L[i] - 1$ and $A[j] \leq a_{i+1}$. Otherwise, $A[i + 1] = A[i]$. We thus need to determine whether such a $j$ exists. This can be done by performing a binary search on the sorted array $L[i]$ to find the largest $j$ such that $L[j] = L[i] - 1$ and $A[j] \leq a_{i+1}$. Note that the array $A$ is decreasing over intervals for which $L$ remains constant, meaning it is sufficient to just check the largest $j$. The time for binary search is $O(\log n)$, so the total update in this case takes $O(\log n)$.

Thus, in either case, we can compute $L[i + 1]$ and $A[i + 1]$ from the values $L[i]$ and $A[i]$ in $O(\log n)$ time. Thus, we can compute $L[n]$ in $O(n \log n)$ time.