Longest Increasing Subsequence
Vassos Hadzilacos

We want to find the length of a longest subsequence of a sequence $A[1..n]$. After examining the problem, we decided that, to solve it, we would solve a related, but slightly different, problem; namely, to compute

$$L(i) = \text{the length of a longest subsequence of } A \text{ that ends in position } i$$

for each $1 \leq i \leq n$. (So, more accurately, we will solve a whole set of related problems.) If we do this, the answer to our original problem is simply $\max\{L(i) : 1 \leq i \leq n\}$.

We then came up with a recursive formula for computing $L(i)$, namely,

$$L(i) = \begin{cases} 
1, & \text{if } A[j] \geq A[i], \text{ for all } 1 \leq j < i \\
1 + \max\{L(j) : 1 \leq j < i \text{ and } A[j] < A[i]\}, & \text{otherwise}
\end{cases} \quad (\dagger)$$

We now need to show that the recursive formula $(\dagger)$ indeed computes $L(i)$ as defined in $(\ast)$. In lecture we effectively did this as we reasoned our way to the dynamic programming algorithm to compute the length of a longest increasing subsequence. Below I give a sample of how you might write up this argument.

**Claim.** The formula $(\dagger)$ correctly computes $L(i)$ as defined in $(\ast)$.

**Proof.** There are two cases:

**Case 1.** For every $j$, $1 \leq j < i$, $A[j] \geq A[i]$. In this case, the longest increasing subsequence of $A$ that ends in position $i$ consists of just $A[i]$, and so it has length 1. So the formula $(\dagger)$ is correct in this case.

**Case 2.** For some $j$, $1 \leq j < i$, $A[j] < A[i]$. Let $S$ be a longest increasing subsequence of $A$ that ends in position $i$. Therefore $S = S' \circ A[i]$, for some sequence $S'$.

$S'$ is an increasing subsequence of $A$ that ends at some $j$, $1 \leq j < i$, such that $A[j] < A[i]$. This is because, otherwise, $S$ would not be an increasing subsequence of $A$ (never mind a longest one).

Furthermore, $S'$ is longest among all longest increasing subsequences of $A$ that end at some position $j$, $1 \leq i < j$, such that $A[j] < A[i]$. For, if $S''$ is not, then there is an increasing subsequence $S''$ of $A$ that ends at some $j$, $1 \leq i < j$, such that $A[j] < A[i]$, and $S''$ is longer than $S'$. But then $S'' \circ A[i]$ is an increasing subsequence of $A$ that ends at $i$ that is longer than $S' \circ A[i] = S$, contradicting the definition of $S$.\(^1\) Therefore, $L(i) = \max\{L(j) : 1 \leq j < i \text{ and } A[j] < A[i]\}$, and the formula $(\dagger)$ is correct in this case. QED

In the presentation of the algorithm in class, we went through the same reasoning in the process of coming up with the formula $(\dagger)$. Here, we simply presented $(\dagger)$ and then proved that it correctly computes $L(i)$.

I don’t much care which alternative you choose when presenting dynamic programming algorithms in your assignments or exams. To some extent it is a matter of taste, and to some extend it depends on the goals of the presentation. When you want to demonstrate the reasoning that led you to the algorithm, the approach I followed in class is perhaps more instructive, but less well organised — as the discovery process usually is! The approach taken here is perhaps clearer but more opaque in terms of demonstrating how one might come up with the algorithm.

---

\(^1\)This is a so-called “cut-and-paste” argument: We “cut” $S'$ and “paste” $S''$. This type of argument is so standard that, in such simple situations, you can simply say “By a cut-and-paste argument...”.