Longest Increasing Subsequence

San Skulrattanakulchai

Feb 29, 2016

Longest Increasing Subsequence

Definition

The sequence $a(i(1)), a(i(2)), \ldots, a(i(\ell))$ is an *increasing subsequence* of the sequence of integers $a(1)$, $a(2)$, $\ldots$, $a(n)$ if $i : [\ell] \to [n]$ is an increasing function and $a(i(j)) < a(i(k))$ whenever $j < k$.

Problem

Given a sequence of distinct integers $a(1)$, $a(2)$, $\ldots$, $a(n)$, find the length of a longest increasing subsequence (LIS). For example, for the sequence 2, 3, 1, 4, the answer is 3 because 3 is the length of the LIS 2, 3, 4.

Solution by Dynamic Programming

For $1 \leq i \leq n$, define $m(i)$ to be the length of any LIS whose last term is $a(i)$.

For example, given the sequence 2, 3, 1, 4, we have $m(1) = 1$, $m(2) = 2$, $m(3) = 1$, and $m(4) = 3$.

Recurrence

For all $1 \leq i \leq n$,

$$m(i) = \max\{1,\ m(j) + 1 : 1 \leq j < i \text{ and } a_j < a_i\}$$

We seek $\max\{m(i) : 1 \leq i \leq n\}$.

Recurrence

For all $1 \leq i \leq n$,

$$m(i) = \max\{1,\ m(j) + 1 : 1 \leq j < i \text{ and } a_j < a_i\}$$

We seek $\max\{m(i) : 1 \leq i \leq n\}$.

If an actual LIS is desired, we can also store the maximizers in a table $z(\cdot)$ while we are filling in the $m(\cdot)$ table. In the last step of the algorithm we use the $z(\cdot)$ table and the input sequence a to retrieve an LIS.