Longest Montone Subsequence Problem

The Problem

We are given a finite sequence $x = \{x_i\}_i^n$ of items from an ordered domain. The goal is to find a monotone increasing sequence $y = \{y_j\}$ of maximal length. For simplicity, we first consider only sequences of unique items. There could be several maximal subsequences, but our algorithm will construct just one of these.

The Algorithm

We first define ℓ_i to be the length of x_i, the greatest length of any monotone subsequence of x whose last term is x_i. If $\ell_i > 1$, we define $\textsc{back}_i$ to be the predecessor of x_i in such a sequence. The choice of this backpointer is also not unique; our algorithm chooses one.

The algorithm creates a ragged matrix of items. The ℓ^{th} column of the matrix contains all items of length ℓ. Each column is doubly sorted: if x_i is below x_j in a column, $i > j$ and $x_i < x_j$.

The algorithm consists of n steps. Initially, we place x_1 at the head of column 1. For all $1 < i \leq n$ we sequentially execute step i.

Step i**.** During this step, ℓ_i is computed, and x_i is placed at the bottom of column ℓ_i. Let L be the number of non-empty columns so far.

(a) If x_i is smaller than the last (bottom) item of column 1, $\ell_i = 1$. Place x_i at the bottom of column 1. In this case, $\textsc{back}_i$ is undefined.

with x_i; $\textsc{back}_i$ is then the last item in column $L - 1$.

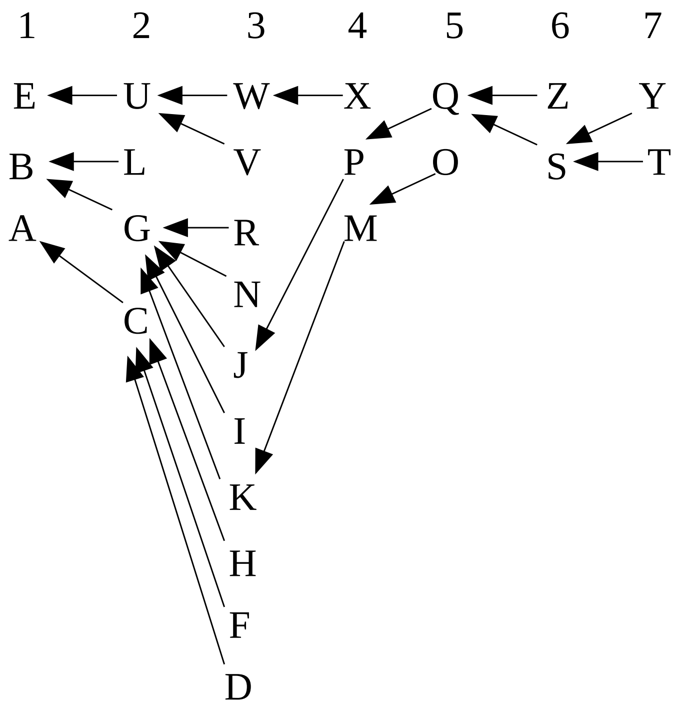
(c) In all other cases, first locate the largest ℓ such that the last item of column ℓ is larger than x_i. We can find this column by binary search, since the last item of column ℓ is less than the last item of column $\ell + 1$ for all ℓ Then insert x_i into that column at the bottom, and define $\textsc{back}_i$ to be the last item of column $\ell - 1$.

Conclusion. Let L be the maximum length of any item, equal to the final number of columns. Pick any item in column L and follow the back pointers from that item to find a monotone increasing subsequence of length L.

Example

We are given a sequence which is a permutation of the Roman alphabet. Find the longest monotone increasing subsequence of σ.

E U W V B L G X R N J P I Q K A M C Z S H O F Y T D



Duplicate Terms

If there are duplicate terms in σ, we can interpret the problem in two ways. To work the algorithm, we use a tiebreaker: if $x_i = x_j$, for $i < j$, which do we consider to be larger?

1. If we require the subsequence to have no duplicates, x_i is considered larger. Thus, x_i and x_j cannot both be in the subsequence.

2. If we allow the subsequence to have duplicates, x_j is considered larger. Thus, x_i and x_j can both be in the subsequence.

For example, if $\sigma =$ D G F F M, then the longest increasing subsequence of σ is either D F M or D F F M.

Stacking Boxes

We are given a set of boxes, each of which has a length, a width, and a height. What is the largest number of boxes that can be stacked? If box 1 is under box 2, then the length and width of box 2 must both be strictly less than the corresponding dimensions of box 1. We are allowed to rotate a box ninety degrees, but only horizontally, since each is marked “This side up” on the top. Explain how the box stacking problem reduces to the longest increasing subsequence problem.