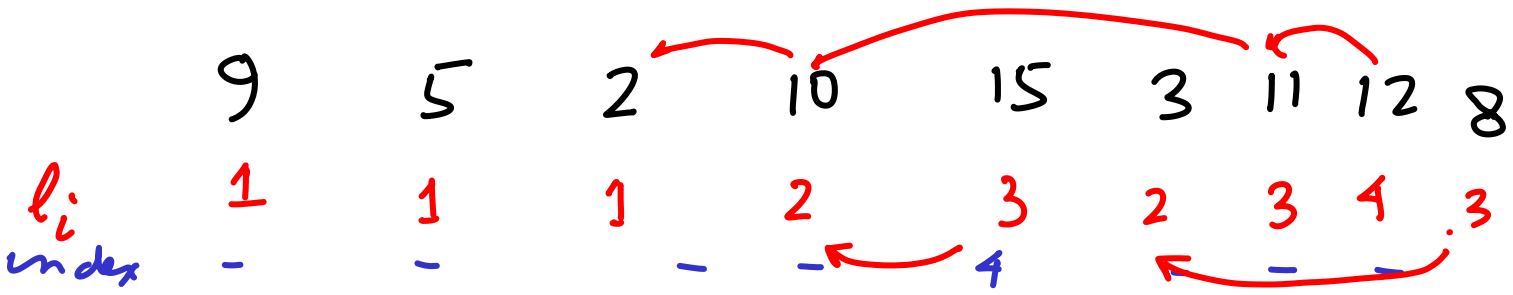
COL 702 Lecture 16b Sept 23

Consider a sequence of nos



If sequence is $x_1\ x_2\ x_3 \ldots x_n$

then $x_{i_1}\ x_{i_2}\ x_{i_3} \ldots x_{i_k}$

is a subsequence for $i_1 < i_2 < i_3 \ldots i_k$

$i_1 : 2 \quad i_2 : 4 \quad i_3 : 5$ in the subsequence $5, 10, 15$

2^n subsequences including empty

If $x_{i_1} < x_{i_2} < x_{i_3} \cdots < x_{i_k}$

- then it is an increasing subsequence

The longest subsequence problem is to find the longest increasing subsequence of a given sequence

Greedy may not work well.

100 200 300 400 1, 2, 3 · · · 99

Dynamic Prog Steps

1. Write a recurrence for the soln : requires a correctness proof
2. Define a tabular version of the subproblems
3. Analyse the running time for filling the table
4. Analyse space

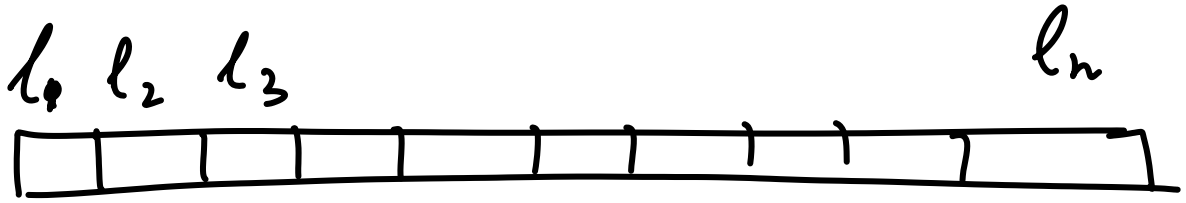
A related problem to the increasing subsequence problem

Find the longest subsequence ending at x_i for all i : call it l_i (is the last term)

Then $\max_i \{l_i\}$ is the overall longest increasing ss

$$l_i = \max_{\substack{j < i \\ x_j < x_i}} \{ l_j + 1 \}$$

$$l(i) = \ldots - l(j)$$



Time to fill entry i : $O(i)$
(max of $i-1$ entry)

$$\sum_{i=1}^{n} O(i) : O(n^2)$$

Space : $O(n)$

Can we do better?

Can be done in $O(n \log n)$ time using Priority Search trees