Consider a sequence of nums

9 5 2 10 15 3 11 12 8

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 \( i_1 < i_2 < i_3 \cdots < i_k \)

\( i_1 = 2 \quad i_2 = 9 \quad i_3 = 5 \) in the subsequence 5, 10, 15

2nd subsequence, including empty

If \( x_{i_1} < x_{i_2} < x_{i_3} \ldots < x_{i_k} \)

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 solution: requires a convinving proof

2. Define a tabular version of the subproblems

3. Analyse the running time for filling the table

4. Analyze space

A related problem to the increasing subsequence problem

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

Then \( \max_i \{ l_i \} \) is the overall longest increasing subsequence.
\[ l_i = \max \left\{ l_j + 1 \right\} \]
\[ j < i \]
\[ x_j < x_i \]

\[ l(i) = \ldots l(j) \]

\[ l_0, l_1, l_2, \ldots, l_n \]

Time to fill entry \( i \) : \( O(i) \)  
\((\text{max } i \text{-th entry})\)

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

Space : \( O(n) \)

Can we do better?

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