

# Example

Below is an example of dynamic programming within the [longest increasing subsequence problem](#).

eg. Given a sequence  $S = \{a_1, a_2, a_3, a_4, \dots, a_{n-1}, a_n\}$ , find the longest subset such that for all  $j$  and  $i$ ,  $j < i$  in the subset  $a_j < a_i$ .

## Thought process

1. Find the value of the longest subsequences (LS<sub>*i*</sub>) at every index  $i$  with last element of sequence being  $a_i$ .
2. Then largest LS<sub>*i*</sub> would be the longest subsequence in the given sequence.

## Writing code

1. Assign LS<sub>*i*</sub> the value of 1 since  $a_i$  is the last element of the sequence.
2. For all  $j$  such that  $j < i$  and  $a_j < a_i$ , find the Largest LS<sub>*j*</sub> and add it to LS<sub>*i*</sub>.
3. This algorithm has a time complexity  $O(n^2)$ .