300: Longest Increasing Subsequence (LIS)

L = [4, 10, 4, 3, 8, 9]

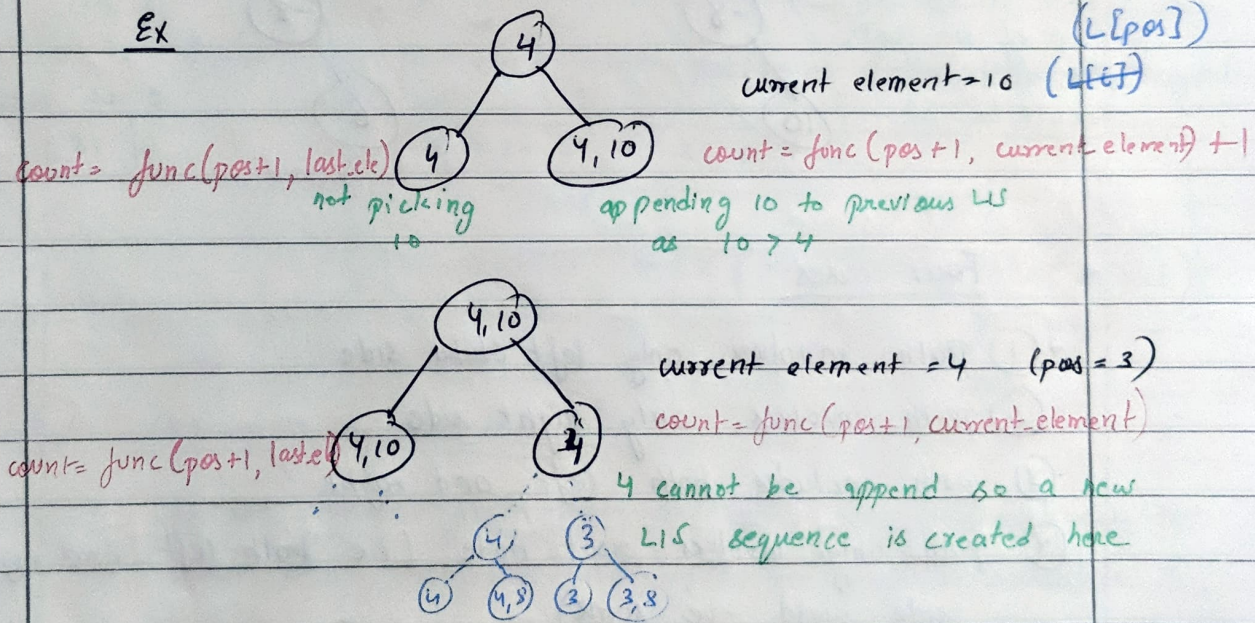
Answers:- [4, 8, 9], [3, 8, 9] ⇒ (3)

Brute force solution

At any position you can pick the element to be part of the LIS (If current element is greater than ~~a~~ maximum element in previous LIS)

Or you can choose not to pick it.

Ex

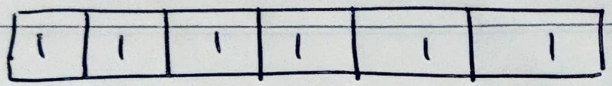


→ Applied the Brute force solution in recursive manner with memorization, but did not work.

```
countA = func(pos+1, last_ele)  ⇒ Don't choose the element
if nums[pos] > last_ele:  (can be appended)
    countB = 1 + func(pos+1, nums[pos])  add count
else:
    countB = func(pos+1, nums[pos])  cannot be appended
return max(countA, countB)
```

(2) Better approach :- Tabulation

4, 10, 4, 3, 8, 9

DP = 

set all to 1, as each pos can individually create on LIS sequence

only one state variable

$DP[i] = \max(DP[j] + 1)$ where $j < i$

Time Complexity : $O(n^2)$

(3) Better approach, not dynamic programming,

This approach gives the right answer, but the LIS it produces is not correct, although the length of the LIS is correct.

Check attached .docx file for this approach.

Time complexity : $O(N \log N)$