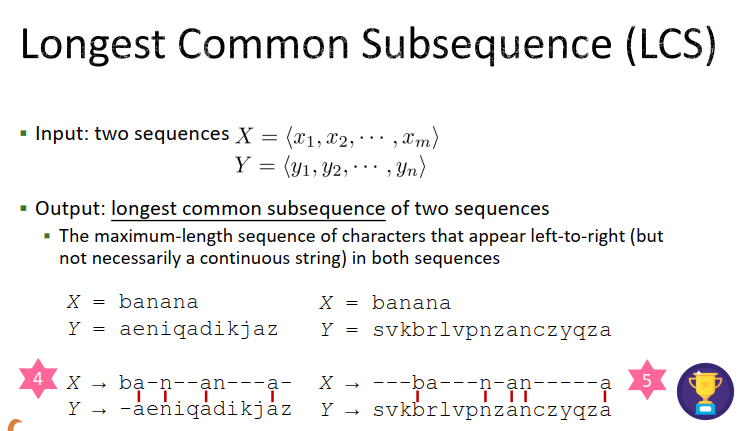
Inclass assignment : Dynamic Programming

**Task: Longest common subsequence**



For example,

X = "ABDE"

Then there can be 16 subsequences of "ABDE"

"",

"A", "B", "D", "E",

"AB", "AD", "AE", "BD", "BE", "DE",

"ABD", "ABE", "ADE", "BDE",

"ABDE".

Y = "ZBE"

Then there can be 8 subsequences of "ZBE"

"",

"Z", "B", "E",

"ZB", "ZE", "BE",

"ZBE".

There are four subsequences that are common in X, Y

Longest of which is "BE", length of which is 2. So answer is 2.

To solve this problem, let us first make some definitions.

X = {x1 x2 …… xm}

Y = {y1 y2 …… yn}

Let us define the i-th prefix of X such that Xi = {x1 x2 …… upto xi}

Similarly the j-th prefix of Y such that Yj = {y1 y2 …… upto yj}

**Recurrence relation to solve the problem:**

LCS(Xi, Yj) = 0 if i or j is 0

LCS(Xi-1, Yj-1) + 1 if xi and yj match

Otherwise max(LCS(Xi-1, Yj), LCS(Xi, Yj-1))