A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters (eg ‘ace’ is a subsequence of ‘abcde ’ while ‘ace’ is not). A common subsequence of two strings is a subsequence that is common in both strings

Ref : LeetCode 1143

Longest common subsequence

text 1 = "abcde"

text 2 = "ace"

X_0 → string starting at 0 i.e $X_0 X_1 \dots X_n$

CASE 1

1st char of text 1 and text 2 match

$$\therefore LCS(X_0, Y_0) = 1 + LCS(X_1, Y_1)$$

CASE 2

1st char of text 1 and text 2 dont match

text 1 = "abcde"

text 2 = "bc"

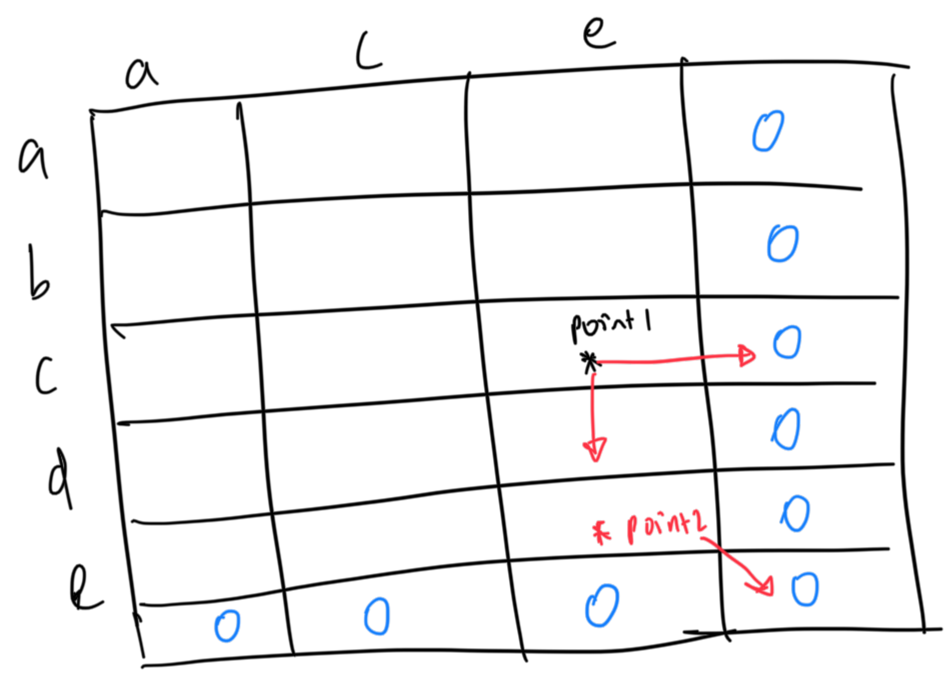
text 1 = $a\boxed{bcde}$

text 2 = $\boxed{bc}$

text 1 = $\boxed{abcde}$

text 2 = $\boxed{\cancel{b}c}$

$$LCS(X_0, Y_0) = MAX\begin{pmatrix} LCS(X_1, Y_0), \\ LCS(X_0, Y_1) \end{pmatrix}$$



* point1

text1 = cde

text 2 = e

val at point 1

CASE 2

MAX $(\downarrow \; \rightarrow)$

* point2

CASE 1

$1 + (\searrow) = 1 + 0 = 1$

```
memo = [ ] [ ]
for i = n to -1    dec -1
    for j = m to -1    dec -1
        if text1[i] = text2[j] :
            memo[i][j] = 1 + memo[i+1][j+1]
        else
        memo[i][j] = MAX [ memo[i+1][j] ,
                           memo[i][j+1] ]
```