* Algorithm of LCS to compute_LCS (A,B) :—

```
m ← leangth (A)
m ← length (B)
for (i=1 to m) do
            c[i,0] ← 0
for (j=0 to n) do
            c[0,j] ← 0
for (i ← 1 to m) do
{
    for (j ← 1 to n) do
        {
            if (ai == bj) then
            {
                    c[i,j] ← c[i-1,j-1]+1
                        d[i,j] ← " ↖ "
            }
            else if (c[i-1,j] ≥ c[i,j-1]) then
            {
                    c[i,j] ← c[i-1,j]
                        d[i,j] ← "↑"
            }
            else
            {
                    c[i,j] ← c[i,j-1]
                        d[i,j] ← "←"
            }
        }
    return c & d
}
```

- To decide the LCS :

```
Algorithm display_LCS (d, A[], i, j)
{
        if ( i=0 || j=0) then
        return
        if ( d[i,j] = " ↖ " then
        {
                display LCS (d, A, i-1, j-1)
                Print (ai)
        }
        else if ( d[i,j] = " ↑ ") then
        {
                display._LCS (d, A, i-1, j )
        }
        else
                display_LCS (d, A, i, j-1)
}.
```

★ Longest Common Subsequence:

$$C[i,j] = \begin{cases} 0 & \text{if } i=0 \ \& \ j=0 \\ C[i-1,j-1]+1 & \text{if } i,j>0 \text{ and } a_i=b_j \\ \max\{C[i,j-1], C[i-1,j]\} & \text{if } i,j>0 \text{ and } \\ & \qquad\qquad a_i \neq b_j. \end{cases}$$

Example: $P = \langle M,N,O,M \rangle$ and
$Q = \langle M,L,N,O,M \rangle$, obtain LCS.

- Initially $i=0$ and $j=0$ $\quad C[i,0] = C[0,j] = 0$.
- compute $C[1,1] \rightarrow a_i = M$ & $b_j = M$. $[a_i = b_j]$
  $$= C[0,0]+1$$
  $$= 0+1 = 1.$$
- compute $C[1,2] \rightarrow a_i = M$ & $b_j = N$ $[a_i \neq b_j]$
  $$= \max\{C[1,1], C[0,2]\}$$
  $$= \max\{1,0\}$$
  $$= 1.$$
- compute $C[1,3] \rightarrow a_i = M$ & $b_j = O$.
  $$= \max\{C[1,2], C[0,3]\}$$
  $$= \max\{1,0\}$$
  $$= 1.$$

⊢ compute $C[1,4] \to a_i = M$ & $b_j = M$.

$$= C[0,3]+1$$
$$= 0+1 = 1.$$

⊢ compute $C[2,1] \to a_i = L$ & $b_j = M$.

$$= \max\{C[2,0], C[1,1]\}$$
$$= \max\{0,1\}$$
$$= 1.$$

⊢ compute $C[2,2] \to a_i = L$ & $b_j = N$.

$$= \max\{C[2,1], C[1,2]\}$$
$$= \max\{1,1\} = 1.$$

⊢ compute $C[2,3] \to a_i = L$ & $b_j = O$

$$= \max\{C[2,2], C[1,3]\}$$
$$= \max\{1,1\} = 1.$$

⊢ compute $C[2,4] \to a_i = L$ & $b_j = M$

$$= \max\{C[2,3], C[1,4]\}$$
$$= \max\{1,1\}$$
$$= 1.$$

⊢ compute $C[3,1] \to a_i = N$ & $b_j = M$

$$= \max\{C[2,1], C[3,0]\}$$
$$= \max\{1,0\}$$
$$= 1.$$

⊢ compute $C[3,2] \to a_i = N$ & $b_j = N$.

$$= C[2,1]+1$$
$$= 1+1$$
$$= 2.$$

⊢ compute $C[3,3] \to a_i = N$ & $b_j = O$

$$= \max\{C[3,2], C[2,3]\}$$
$$= \max\{2,1\}$$
$$= 2.$$

⊢ compute $C[3,4] \to a_i = N$ & $b_j = M$.

$$= \max\{C[3,3], C[2,4]\}$$
$$= \max\{2,1\}$$
$$= 2.$$

⊢ compute $c[4,1]$ → $ai = 0$ & $bj = M$.

$\quad = max\{c[4,0], c[3,1]\}$

$\quad = max\{0,1\}$

$\quad = 1$.

⊢ compute $c[4,2]$ → $ai = 0$ & $bj = N$

$\quad = max\{c[4,1], c[3,2]\}$

$\quad = max\{1,2\}$

$\quad = 2$.

⊢ compute $c[4,3]$ → $ai = 0$ & $bj = 0$

$\quad = c[3,2]+1$

$\quad = 2+1$

$\quad = 3$.

⊢ compute $c[4,4]$ → $ai = 0$ & $bj = M$.

$\quad = max\{c[4,3], c[3,4]\}$

$\quad = max\{3,2\}$

$\quad = 3$.

⊢ compute $c[5,1]$ → $ai = M$ & $bj = M$

$\quad = c[4,0]+1$

$\quad = 0+1$

$\quad = 1$.

⊢ compute $c[5,2]$ → $ai = M$ & $bj = N$

$\quad = max\{c[5,1], c[4,2]\}$

$\quad = max\{1,2\}$

$\quad = 2$.

⊢ compute $c[5,3]$ → $ai = M$ & $bj = 0$

$\quad = Max\{c[5,2], c[4,3]\}$

$\quad = Max\{2,3\}$

$\quad = 3$

⊢ compute $c[5,4]$ → $ai = M$ & $bj = M$

$\quad = c[4,3]+1$

$\quad = 3+1 = 4$.

$\quad$ SO, LCS = MNOM.