

LCS Algorithm

LCS-LENGTH(X, Y, m, n)

```
1  let  $b[1:m, 1:n]$  and  $c[0:m, 0:n]$  be new tables
2  for  $i = 1$  to  $m$ 
3       $c[i, 0] = 0$ 
4  for  $j = 0$  to  $n$ 
5       $c[0, j] = 0$ 
6  for  $i = 1$  to  $m$           // compute table entries in row-major order
7      for  $j = 1$  to  $n$ 
8          if  $x_i == y_j$ 
9               $c[i, j] = c[i - 1, j - 1] + 1$ 
10              $b[i, j] = \nwarrow$ 
11         elseif  $c[i - 1, j] \geq c[i, j - 1]$ 
12              $c[i, j] = c[i - 1, j]$ 
13              $b[i, j] = \uparrow$ 
14         else  $c[i, j] = c[i, j - 1]$ 
15              $b[i, j] = \leftarrow$ 
16  return  $c$  and  $b$ 
```

PRINT-LCS(b, X, i, j)

```
1  if  $i == 0$  or  $j == 0$ 
2      return          // the LCS has length 0
3  if  $b[i, j] == \nwarrow$ 
4      PRINT-LCS( $b, X, i - 1, j - 1$ )
5      print  $x_i$       // same as  $y_j$ 
6  elseif  $b[i, j] == \uparrow$ 
7      PRINT-LCS( $b, X, i - 1, j$ )
8  else PRINT-LCS( $b, X, i, j - 1$ )
```