Najdłuższy Wspólny Podciąg (Longest Common Subsequence)

Jest to przykład algorytmu ilustrującego metodę *programowania dynamicznego*.

Najdłuższy Wspólny Podciąg (wersja iteracyjna)

Ciągi wejściowe indeksowane od pozycji `1` , czyli
`x[1], x[2], ... , x[m]`
`y[1], y[2], ... , y[n]` .
Tablice `c, b` indeksowane od `0` .

```
LCS-Length(x,y)
  m = x.length
  n = y.length
  for i = 0 to m
    c[i,0] = 0
  for j = 1 to n
    c[0,j] = 0
  for i = 1 to m
    for j = 1 to n
      if x[i]==y[j]
         c[i,j] = c[i-1,j-1]+1
         b[i,j] = "\"
      else if c[i-1,j] >= c[i,j-1]
              c[i,j] = c[i-1,j]
              b[i,j] = "|"
           else
              c[i,j] = c[i,j-1]
              b[i,j] = "-"
  return b,c
```

```
PrintLCS(x,y,b,i,j)
  if i==0 lub j==0
      return
  if b[i,j] == "\"
     PrintLCS(X,Y,b,i-1,j-1)
     drukuj x[i]
  else if b[i,j] == "|"
          PrintLCS(X,Y,b,i-1,j)
       else
          PrintLCS(X,Y,b,i,j-1)
```

Wersja rekurencyjna ze spamiętywaniem

```
LCS-Rec(x,y)
  m = x.length
  n = y.length
  for i <- 1 to m
    for j <- 1 to n
        c[i,j]= INFINITY
  for i = 1 to m
     c[i,0] = 0
  for j = 0 to n
     c[0,j] = 0
  LCS-Lookup(m,n,c,b)
  return b,c

LCS-Lookup(i,j,c,b)
  if c[i,j]<INFINITY
     return c[i,j]
  if x[i]=y[j]
     c[i,j] = LCS-Lookup(i-1,j-1,c,b)+1
     b[i,j] = "\"
   else c1 = LCS-Lookup(i-1,j,c,b)
        c2 = LCS-Lookup(i,j-1,c,b)
        if c1>=c2 // czyli c[i-1,j] >= c[i,j-1]
           c[i,j] = c1
           b[i,j] = "|"
        else
            c[i,j] = c2
            b[i,j] = "-"
  return c[i,j]
```