



ALGORİTMA ANALİZİ 4. ÖDEV DİNAMİK PROGRAMLAMA

Adı Soyadı: Büşra Medine GÜRAL

Öğrenci Numarası: 20011038

Mail: medine.gural@std.yildiz.edu.tr

Dersin Öğretmeni: M. Elif KARSLIGİL

Video Linki: -

1. Problem Tanımı

İlgili ödevde, kullanıcıdan alınan iki farklı uzunluktaki string üzerinde, dinamik programlama yaklaşımı ile en uzun ortak alt dizinin bulunması istenmektedir. İlk olarak, kullanıcıdan iki adet string bilgisi alınacak ve dinamik programlama matrisi oluşturularak LCS algoritması çalıştırılacaktır. En uzun sekansa sahip kelimelerin bulunabilmesi için bir kontrol matrisi tutulacaktır. Matrisler tekrarlı adımlarda ekrana yazdırılacaktır. Bu süreç tamamlandığında en uzun sekansın boyutu ve eğer birden fazla ise bu sekanslar ekrana yazdırılmalıdır.

2. Problem Çözümü

Öncelikle, kullanıcıdan girilecek kelimelerin uzunluğu ve kelimeler sırasıyla alınmaktadır. Alınan kelimelerin uzunluğuna göre iki ayrı 0 değerli matrisler oluşturulmaktadır. İlk matris sekansın uzunluğunu hesaplamak için, ikinci matris ise oluşan sekansın hangi kelime olduğunu ekrana yazdırabilmek için kullanılmaktadır.

İlk matriste (1,1) noktasından başlanarak bir önceki adım kontrol edilmektedir. Bulunulan hücreye denk gelen harfler eşit ise ilgili hücrenin sol üst köşegeninde bulunan sayının bir fazlası o hücreye kaydedilmektedir. Kontrol matrisinde ise aynı hücre '1' sayısı ile işaretlenmektedir. Eğer hücreye denk gelen harfler eşit değilse, ilgili hücrenin bir üst ve bir sol hücreleri kontrol edilerek en büyük sayı alınmaktadır. Bu durumdaki kontrol matrisinde ilgili hücre; üstteki sayı büyükse '2', soldaki sayı büyükse '3' rakamını almaktadır. Son koşul olarak harflerin aynı olmadığı ve bir sol hücre ile bir üst hücrenin eşit olduğu durumda dinamik olarak oluşturulan matris için soldan gelen değer kabul edilerek ilerlenilmektedir. Kontrol matrisinde ise hücre '4' ile işaretlenmektedir. Bu şekilde oluşturulan dinamik matris için en son hücre, en uzun ortak sekansın boyutunu belirtmektedir.

Kontrol matrisinde, en son hücreden ilk hücreye doğru rekürsif bir şekilde gidilerek en uzun sekans ya da sekanslar ekrana yazdırılmaktadır. İşaretlenen sayılara göre ayrılan koşullar bulunmaktadır. Örneğin ilgili hücre 1 sayısına sahipse bu hücreye köşegen üzerinden gelindiği, 2 ise hücreye üst taraftan gelindiği anlaşılmaktadır. Algoritmanın kilit noktası, kontrol matrisindeki hücrelerin 4 ile işaretlendiği hücrelerdir. 4 sayısı, soldan ya da üstten devam edilebileceği anlamına geldiğinden iki taraftan rekürsif işlemler gerçekleştirilmektedir, böylelikle birden fazla en uzun ortak sekans bulunduğunda kullanıcıya gösterilebilecektir. Başlangıç satırına/sütununa dönüldüğünde ise bulunan kelimenin daha önceden ekrana yazdırılıp yazdırılmadığı kontrol edilmektedir. Bunun için ayrı bir liste tutulmakta ve liste sekanslara bağlı olarak sürekli güncellenmektedir.

Algoritmanın sonucunda bulunan en uzun ortak kelimeler ve uzunluğu kullanıcıya gösterilmektedir.

3. Karşılaşılan Sorunlar

Yaşanılan tek büyük problem, birden fazla ortak sekansın olduğu durumda nasıl bir yol izleneceğine dair düşünülmesiydi. Bir üst hücre ve bir sol hücrenin aynı değere sahip olması kontrol matrisini etkilediğinden rekürsif yapının nasıl olması gerektiği ile ilgili sorunlar yaşanılmıştır. Ardından parametre güncellemeleri ile bu sorun halledilmiş fakat bu sefer de aynı kelimelerin tekrar tekrar yazdırılması sorunu ortaya çıkmıştır. Bu sorun ise kullanılan ek parametreler ve bulunan kelimenin önceki kelimelerle karşılaştırılması yoluyla ortadan kalkmıştır.

4. Karmaşıklık Analizi

Koddaki her bir fonksiyon için karmaşıklık incelendiğinde:

- printMatrix fonksiyonu, matrisleri ekrana yazdırmak için tüm matrisi dolaştığından karmaşıklığı $O(n^2)$ 'dir.
- isSequencePrinted fonksiyonu, bir dizi üzerinde eşleşen kelime olup olmadığını aradığından karmaşıklığı $O(n)$ olmaktadır.

- addPrintedSequence fonksiyonu, bir kelimeyi bir diziye doğrudan eklediğinden sabit sayılı işlem yapmaktadır ve karmaşıklığı $O(1)$ 'dir.
- findAllLCS fonksiyonu, en uzun tüm ortak sekansları bulmaktadır ve karmaşıklığı, sondan başa doğru dallanan i ve j değerlerine bağlıdır. Bu i ve j değerleri ise matrisin son elemanından itibaren başladığından ve ikiye bölünerek ilerlediğinden karmaşıklık $O(2^{(m+n)})$ olarak ifade edilir. Ek olarak, fonksiyon içerisinde yapılan kelime tekrarı kontrolü seqCount kadarlık bir karmaşıklık daha eklemekte ve sonuçtaki karmaşıklık $O(2^{(m+n)*k})$ olmaktadır.
- lcs fonksiyonu, en uzun ortak sekansın boyutunu bulup findAllLCS fonksiyonunu çalıştırmaktadır ve karmaşıklığı şu şekildedir:
 - Matris oluşturmak için bir döngüye sahiptir. $\rightarrow O(n)$
 - Matrislerin değerlerini oluşturmak için iç içe iki döngüye sahiptir. $\rightarrow O(nm)$
 - findAllLCS fonksiyonunu çağırır. $\rightarrow O(2^{(m+n)*k})$
 - Bellek temizlemek için kullanılan döngü bulunmaktadır. $\rightarrow O(n)$

$O(n + n + n^2 + 2^{(m+n)*k}) = O(2^{(m+n)*k})$ olmaktadır.

$$lcs[i, j] = \begin{cases} \emptyset & , i = \emptyset \text{ ya da } j = \emptyset \\ lcs[i-1][j-1] + 1 & , x_i = y_j \\ \max(lcs[i][j-1], lcs[i-1][j]) & , \text{otherwise} \end{cases}$$

Yazılan kodun tamamı için karmaşıklık hesabı yapıldığında, en fazla karmaşıklık değerine sahip fonksiyon seçilmelidir, bu sebeple karmaşıklık $O(2^{(m+n)*k})$ olmaktadır.

lcs

input:

firstWord – the first word recieved by user
 secondWord – the second word recieved by user
 firstLength – length of first word
 secondLength – length of second word

```
function lcs(firstWord, secondWord, firstLength, secondLength)
  LCSMatrix = createMatrix(firstLength+1, secondLength+1)
  controlMatrix = createMatrix(firstLength+1, secondLength+1)
  printedSeq = createArray(SEQ_COUNT)
  seqCount = 0
  for i from 0 to firstLength do
    for j from 0 to secondLength do
      if firstWord[i-1] == secondWord[j-1] then
        LCSMatrix[i][j] = LCSMatrix[i-1][j-1] + 1
        controlMatrix[i][j] = 1 // The same letters, came from the diagonal
      else
        if LCSMatrix[i-1][j] > LCSMatrix[i][j-1] then
          LCSMatrix[i][j] = LCSMatrix[i-1][j]
          controlMatrix[i][j] = 2 // The number above is larger, came from the top row
        else if LCSMatrix[i-1][j] < LCSMatrix[i][j-1] then
          LCSMatrix[i][j] = LCSMatrix[i][j-1]
          controlMatrix[i][j] = 3 // The number left is larger, came from the left column
```

```

        else
            LCSMatrix[i][j] = LCSMatrix[i][j-1]
            controlMatrix[i][j] = 4 // Top and bottom numbers are equal
        end if
    end if
end for
end for
findAllLCS(firstWord, secondWord, controlMatrix, firstLength, secondLength, currentLCS,
LCSMatrix[firstLength][secondLength] - 1, printedSeq, seqCount)
return LCSMatrix[firstLength][secondLength]
end function

```

findAllLCS

input:

firstWord – the first word recieved by user
 secondWord – the second word recieved by user
 controlMatrix - the matrix storing the path of backtracked Word
 i – current indices in the matrix
 j – current indices in the matrix
 currentLCS – current LCS sequence
 currentIndex – current index in the LCS sequence
 printedSeq – array of already printed sequences
 seqCount - pointer to the number of sequences in the array

```

function findAllLCS(firstWord, secondWord, controlMatrix, i, j, currentLCS, currentIndex, printedSeq,
seqCount)
    if i == 0 or j == 0 then
        if not isSequencePrinted(currentLCS, printedSeq, seqCount) then
            print("LCS: {}", currentLCS)
            addPrintedSequence(currentLCS, printedSeq, seqCount)
        end if
        return
    end if
    if controlMatrix[i][j] == 1 then
        currentLCS[currentIndex] = firstWord[i - 1]
        findAllLCS(firstWord, secondWord, controlMatrix, i - 1, j - 1, currentLCS, currentIndex - 1,
printedSeq, seqCount)
    else
        if controlMatrix[i][j] == 2 or controlMatrix[i][j] == 4 then
            findAllLCS(firstWord, secondWord, controlMatrix, i - 1, j, currentLCS, currentIndex, printedSeq,
seqCount)
        end if
        if controlMatrix[i][j] == 3 or controlMatrix[i][j] == 4 then
            findAllLCS(firstWord, secondWord, controlMatrix, i, j - 1, currentLCS, currentIndex, printedSeq,
seqCount)
        end if
    end if
end function

```

5. Ekran Görüntüleri

Örnek 1: İletilen dokümanın ilk örneği

1. kısım

```
Please enter the length of the first word: 6
Please enter the first word. abcdzb
Please enter the length of the second word: 7
Please enter the second word. bacedab

DYNAMIC MATRIX
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

CONTROL MATRIX
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0

DYNAMIC MATRIX
0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

2. kısım

```
CONTROL MATRIX
0 0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

DYNAMIC MATRIX
0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 2
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

CONTROL MATRIX
0 0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 1 4 4 4 4 4 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

DYNAMIC MATRIX
0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 2
0 1 1 2 2 2 2 2
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

CONTROL MATRIX

```

0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 1 4 4 4 4 4 1
0 2 4 1 3 3 3 4
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 2
0 1 1 2 2 2 2 2
0 1 1 2 2 3 3 3
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

CONTROL MATRIX

```

0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 1 4 4 4 4 4 1
0 2 4 1 3 3 3 4
0 2 4 2 4 1 3 3
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 2
0 1 1 2 2 2 2 2
0 1 1 2 2 3 3 3
0 1 1 2 2 3 3 3
0 0 0 0 0 0 0 0

```

CONTROL MATRIX

```

0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 1 4 4 4 4 4 1
0 2 4 1 3 3 3 4
0 2 4 2 4 1 3 3
0 2 4 2 4 2 4 4
0 0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1
0 1 1 1 1 1 1 2
0 1 1 2 2 2 2 2
0 1 1 2 2 3 3 3
0 1 1 2 2 3 3 3
0 1 1 2 2 3 3 4

```

CONTROL MATRIX

```

0 0 0 0 0 0 0 0
0 4 1 3 3 3 1 3
0 1 4 4 4 4 4 1
0 2 4 1 3 3 3 4
0 2 4 2 4 1 3 3
0 2 4 2 4 2 4 4
0 1 4 2 4 2 4 1

```

LCS: acdb

LCS: bcdb

Length of LCS: 4

Örnek 2: Aynı iki kelime

1.kısım

```
Please enter the length of the first word: 3
Please enter the first word. qwe
Please enter the length of the second word: 3
Please enter the second word. qwe

DYNAMIC MATRIX

0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

CONTROL MATRIX

0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0

DYNAMIC MATRIX

0 0 0 0
0 1 1 1
0 0 0 0
0 0 0 0

CONTROL MATRIX

0 0 0 0
0 1 3 3
0 0 0 0
0 0 0 0
```

2.kısım

DYNAMIC MATRIX

```
0 0 0 0
0 1 1 1
0 1 2 2
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 1 3 3
0 2 1 3
0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0
0 1 1 1
0 1 2 2
0 1 2 3
```

CONTROL MATRIX

```
0 0 0 0
0 1 3 3
0 2 1 3
0 2 2 1
```

LCS: qwe

Length of LCS: 3

Örnek 3: Eşleşme olmayan iki kelime

1.kısım

```
Please enter the length of the first word: 4
Please enter the first word. test
Please enter the length of the second word: 3
Please enter the second word. yum
```

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 4 4 4
0 0 0 0
0 0 0 0
0 0 0 0
```

2. kısım

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 4 4 4
0 4 4 4
0 0 0 0
0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 4 4 4
0 4 4 4
0 4 4 4
0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```


3. kısım

DYNAMIC MATRIX

```
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0
0 4 4 4
0 4 4 4
0 4 4 4
0 4 4 4
```

LCS:

Length of LCS: 0

Örnek 4: Tekrarlı harfler içeren kelimeler

1.kısım

```
Please enter the length of the first word: 6
Please enter the first word. zzbbcc
Please enter the length of the second word: 6
Please enter the second word. zbczbc
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

2.kısım

CONTROL MATRIX

```
0 0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0
0 1 1 1 1 1 1
0 1 1 1 2 2 2
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 1 4 4 1 3 3
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0
0 1 1 1 1 1 1
0 1 1 1 2 2 2
0 1 2 2 2 3 3
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

CONTROL MATRIX

```

0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 1 4 4 1 3 3
0 2 1 3 4 1 3
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0
0 1 1 1 1 1 1
0 1 1 1 2 2 2
0 1 2 2 2 3 3
0 1 2 2 2 3 3
0 0 0 0 0 0 0
0 0 0 0 0 0 0

```

CONTROL MATRIX

```

0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 1 4 4 1 3 3
0 2 1 3 4 1 3
0 2 1 4 4 1 4
0 0 0 0 0 0 0
0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0
0 1 1 1 1 1 1
0 1 1 1 2 2 2
0 1 2 2 2 3 3
0 1 2 2 2 3 3
0 1 2 3 3 3 4
0 0 0 0 0 0 0

```

CONTROL MATRIX

```

0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 1 4 4 1 3 3
0 2 1 3 4 1 3
0 2 1 4 4 1 4
0 2 2 1 3 4 1
0 0 0 0 0 0 0

```

DYNAMIC MATRIX

```

0 0 0 0 0 0 0
0 1 1 1 1 1 1
0 1 1 1 2 2 2
0 1 2 2 2 3 3
0 1 2 2 2 3 3
0 1 2 3 3 3 4
0 1 2 3 3 3 4

```

CONTROL MATRIX

```

0 0 0 0 0 0 0
0 1 3 3 1 3 3
0 1 4 4 1 3 3
0 2 1 3 4 1 3
0 2 1 4 4 1 4
0 2 2 1 3 4 1
0 2 2 1 4 4 1

```

LCS: zzbc

LCS: zbbc

LCS: zbcc

Length of LCS: 4

Örnek 5: İletilen dokümanın 2. örneği

Matris boyutlarından dolayı aradaki aşamaların ekran görüntüleri raporda değil, videoda gösterilmiştir.

```
Please enter the length of the first word: 13
Please enter the first word. maerbphcappba
Please enter the length of the second word: 13
Please enter the second word. amrrerchazbza
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

CONTROL MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 2 2 2 2 2 2
0 1 1 1 1 2 2 2 2 2 2 2 2 2 2
0 1 1 2 2 2 3 3 3 3 3 3 3 3 3
0 1 1 2 2 2 3 3 3 3 3 4 4 4 4
0 1 1 2 2 2 3 3 3 3 3 4 4 4 4
0 1 1 2 2 2 3 4 4 4 4 4 4 4 4
0 1 1 2 2 2 3 4 4 5 5 5 5 5 5
0 1 1 2 2 2 3 4 4 5 5 5 5 5 5
0 1 1 2 2 2 3 4 4 5 5 5 5 5 5
0 1 1 2 2 2 3 4 4 5 5 6 6 6 6
0 1 1 2 2 2 3 4 4 5 5 6 6 7
```

CONTROL MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4 1 3 3 3 3 3 3 3 3 3 3 3 3
0 1 4 4 4 4 4 4 4 1 3 3 3 3 1
0 2 4 4 4 1 3 3 3 4 4 4 4 4 4
0 2 4 1 1 4 1 3 3 3 3 3 3 3 3
0 2 4 2 4 4 2 4 4 4 4 1 3 3 3
0 2 4 2 4 4 2 4 4 4 4 2 4 4 4
0 2 4 2 4 4 2 4 1 3 3 4 4 4 4
0 2 4 2 4 4 2 1 4 4 4 4 4 4 4
0 1 4 2 4 4 2 2 4 1 3 3 3 3 1
0 2 4 2 4 4 2 2 4 2 4 4 4 4 4
0 2 4 2 4 4 2 2 4 2 4 4 4 4 4
0 2 4 2 4 4 2 2 4 2 4 1 3 3 3
0 1 4 2 4 4 2 2 4 1 4 2 4 1
```

LCS: merhaba

LCS: aerhaba

LCS: mercaba

LCS: aercaba

Length of LCS: 7

Örnek 6: İletilen dokümanın son örneği

Matris boyutlarından dolayı aradaki aşamaların ekran görüntüleri raporda değil, videoda gösterilmiştir.

```
Please enter the length of the first word: 11
Please enter the first word. rsbecgreeat
Please enter the length of the second word: 12
Please enter the second word. kreslgecrfet

DYNAMIC MATRIX

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

CONTROL MATRIX

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

DYNAMIC MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 1 1 2 2 2 2 2 2 2 2 2 2 2
0 0 1 1 2 2 2 2 2 2 2 2 2 2 2
0 0 1 2 2 2 2 3 3 3 3 3 3 3 3
0 0 1 2 2 2 2 3 4 4 4 4 4 4 4
0 0 1 2 2 2 3 3 4 4 4 4 4 4 4
0 0 1 2 2 2 3 3 4 5 5 5 5 5 5
0 0 1 2 2 2 3 4 4 5 5 6 6 6 6
0 0 1 2 2 2 3 4 4 5 5 6 6 6 6
0 0 1 2 2 2 3 4 4 5 5 6 6 6 6
0 0 1 2 2 2 3 4 4 5 5 6 7
```

CONTROL MATRIX

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 4 1 3 3 3 3 3 3 1 3 3 3 3 3
0 4 2 4 1 3 3 3 3 3 3 3 3 3 3
0 4 2 4 2 4 4 4 4 4 4 4 4 4 4
0 4 2 1 4 4 4 1 3 3 3 1 3 3 3
0 4 2 2 4 4 4 2 1 3 3 3 3 3 3
0 4 2 2 4 4 1 4 2 4 4 4 4 4 4
0 4 1 2 4 4 2 4 2 1 3 3 3 3 3
0 4 2 1 4 4 2 1 4 2 4 1 3 3 3
0 4 2 1 4 4 2 1 4 2 4 1 4 4 4
0 4 2 2 4 4 2 2 4 2 4 2 4 4 4
0 4 2 2 4 4 2 2 4 2 4 2 1 4 4
```

LCS: rsecret

Length of LCS: 7