TÜRKİYE CUMHURİYETİ

YILDIZ TEKNİK ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Algoritma Analizi Dersi 4. Ödev Raporu

Öğrenci No: 21011084

Ad-Soyad: Berkay Gümüşay

E-Posta: berkay.gumusay@std.yildiz.edu.tr

Telefon: 0 535 840 79 78

Video Linki: https://www.youtube.com/watch?v=uSeDKtYROyo

Raporun İçerikleri

Problemin Tanımı

Bizden verilen iki karakter dizisi içerisindeki aynı uzunluğa sahip ortak ifadeyi/ifadeleri bulan bir program yazmamız istenmiştir.

Problemin Çözümü

İstenen bu problem dinamik programlama yardımıyla çözülebilir.

Kullanıcıdan ilk başta iki karakter dizisi girdi olarak alınır. Daha sonrasında alınan iki karakter dizisi kullanılarak bir matris oluşturulup. Dinamik programlama kurallarına uyularak bu matris doldurulur. Eğer iki karakter aynı ise bir sol-üstteki değer 1 arttırılarak lcsMatrix'in mevcut indisine yazılır ve pickedMatrix'in mevcut indisindeki değer 1 yapılır. Eğer karakter farklı ise lcsMatrix'te bir soldaki değer ile bir üstteki değerden büyük olan mevcut indise indise yazılır, pickedMatrix'te ise mevcut indise 0 yazılır.

Matris doldurma işlemi bittikten sonra MxN'lik matrisin M. Satır N. sütundaki son elemanı en uzun ifadenin uzunluğu olmuş olur. Sonrasında bu indisteki elemandan başlanarak yukarıya doğru backtracking işlemi yapılarak en uzun ortak ifade/ifadeler bulunur ve ekrana yazdırılır.

Backtracking işlemi ise lcsMatrix'in M. satır N. sütunundaki elemandan başlayıp sol üst çaprazdaki(M-1, N-1) değerin artıp artmadığını kontrol ederek öncesinde seçilmiş tüm karakter dizilerini recursive bir şekilde bulur.

Karşılaşılan Sorunlar

Problemin çözümünde çoğu kısım derste gösterildiği için pek fazla sorun ile karşılaşmadım. Karşılaştığım sorunlar genellikle aynı uzunluğa sahip ortak elemanlar birden fazlayken hepsini bulma kısmındaydı.

Karmaşıklık Analizi

a -) Boş LCS Matrisini Dolduran Fonksiyon:

LCS Matrisini doldurma işleminin yapıldığı bölgenin sözde kodu:

```
for i from 0 to length(stringX) do:
   for j from 0 to length(stringY) do:
      if i = 0 or j = 0 then:
            lcsMatrix[i][j] <- 0
            pickedMatrix[i][j] <- 0
      else if stringX[i-1] == string = stringY[j-1] then:
            lcsMatrix[i][j] <- lcsMatrix[i-1][j-1] + 1
            pickedMatrix[i][j] <- 1
      else:
            set lcsMatrix[i][j] <- max( lcsMatrix[i-1][j], lcsMatrix[i][j-1] )
            pickedMatrix[i][j] <- 0
      end if
   end for
end for
```

İlk karakter dizisinin uzunluğu M, ikinci karakter dizisinin uzunluğu N olduğu bir durumda açtığımız matrisin boyutu MxN olacaktır. Ve bu matrisin her indisi gezilerek işlemler yapıldığından dolayı MxN kez işlem yapılacaktır. Bu yüzden zaman karışıklığına **O(MxN)** diyebiliriz.

b -) Backtracking İşlemi Yaparak Ortak İfadeleri Bulan Fonksiyon:

Ortak ifadeleri bulma işleminin yapıldığı bölgenin sözde kodu:

```
trackSubsequences(stringX, stringY, lcsMatrix, tmpLCS, lcsIndex, uniqueSubSeqs, subSeqCount, rows, cols):
   if rows = 0 || cols = 0 then:
      if isSubsequenceUnique(uniqueSubSeqs, subSeqCount, tmpLCS) then:
         uniqueSubSeqs[subSeqCount] <- tmpLCS
         subSeqCount++
      return

   if stringX[rows - 1] = stringY[cols - 1] then:
      tmpLCS[--lcsIndex] <- stringX[rows - 1]
      trackSubsequences(stringX, stringY, lcsMatrix, tmpLCS, lcsIndex, uniqueSubSeqs, subSeqCount, rows - 1, cols - 1)
      lcsIndex++
   else:
      if lcsMatrix[rows - 1][cols] >= lcsMatrix[rows][cols - 1] then:
         trackSubsequences(stringX, stringY, lcsMatrix, tmpLCS, lcsIndex, uniqueSubSeqs, subSeqCount, rows - 1, cols)
      if lcsMatrix[rows][cols - 1] >= lcsMatrix[rows - 1][cols] then:
         trackSubsequences(stringX, stringY, lcsMatrix, tmpLCS, lcsIndex, uniqueSubSeqs, subSeqCount, rows, cols - 1)
```

İlk karakter dizisinin uzunluğu M, ikinci karakter dizisinin uzunluğu N olduğu bir durumda açtığımız matrisin boyutu MxN olacaktır. Yukarıdaki kodun en kötü durumunda her indis için 2 tane rekürsif çağır yapılacağını düşünürsek zaman karmaşıklığı **O(2^(m+n))** olacaktır.

Ekran Çıktıları

Örnek 1:

String1 : BDCB | String2 : BADCB

Ekran Çıktısı:

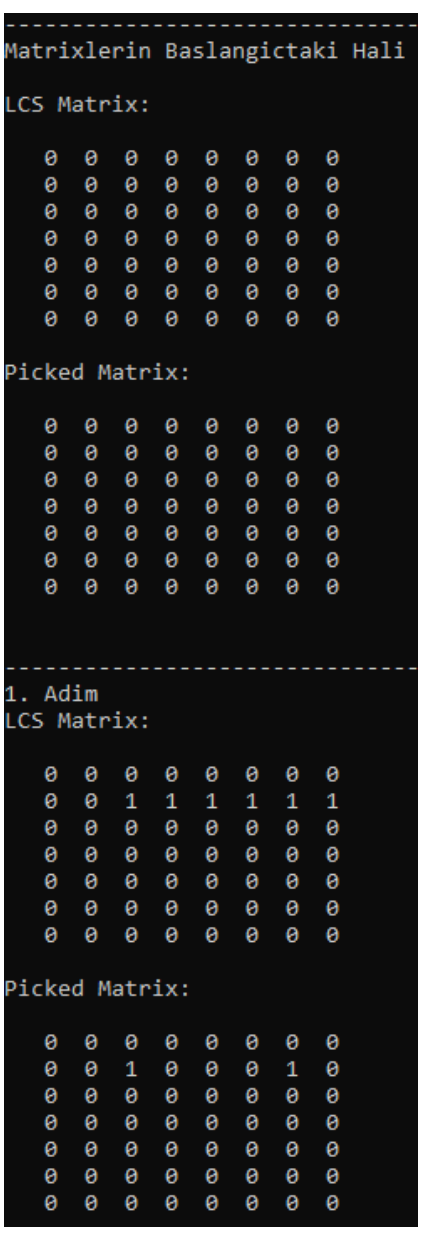
```
------------------------------------------------------------------
Matrixlerin Baslangictaki Hali

LCS 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


Picked 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



------------------------------------------------------------------
1. Adim
LCS Matrix:

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


Picked Matrix:

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



------------------------------------------------------------------
2. Adim
LCS Matrix:

   0  0  0  0  0  0
   0  1  1  1  1  1
   0  1  1  2  2  2
   0  0  0  0  0  0
   0  0  0  0  0  0


Picked Matrix:

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

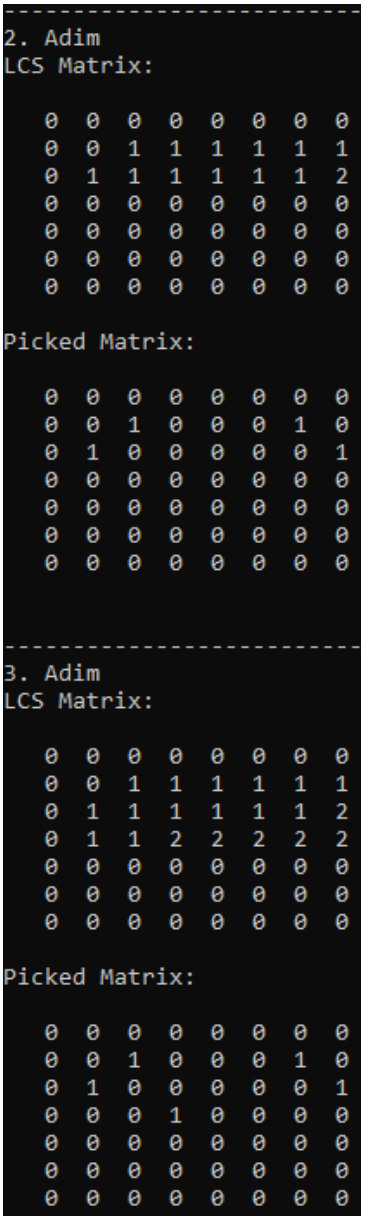
```
------------------------------------------------------------------
3. Adim
LCS Matrix:

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


Picked Matrix:

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



------------------------------------------------------------------
4. Adim
LCS Matrix:

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


Picked Matrix:

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



------------------------------------------------------------------
Word1: BDCB
Word2: BADCB
Length of LCS: 4
All LCS:
BDCB
```

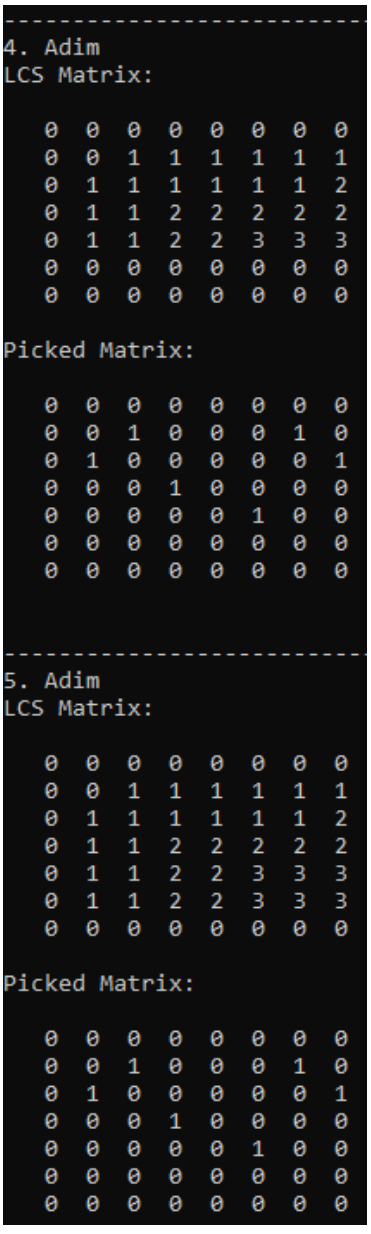
Örnek 2:

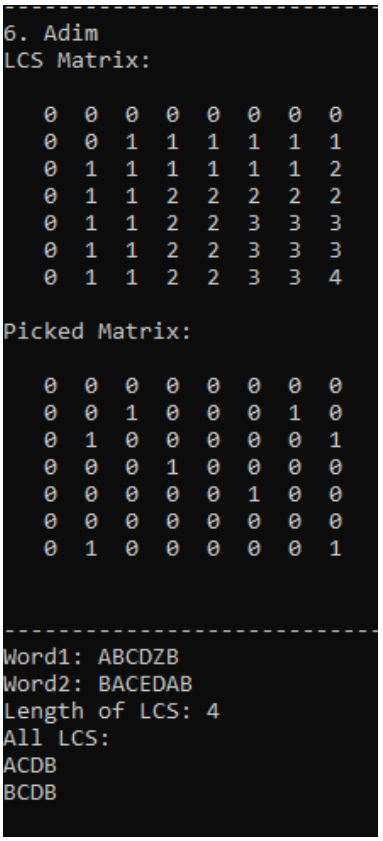
String1: BACEDAB | String2: ABCDZB

Ekran Çıktısı:









(Önceki örneklerde matrisler gösterildiğinden dolayı bundan sonraki örneklerde yer kaplamaması açısından matrisler eklenmemiştir.)

Örnek 3:

String1: MAERBPHCAPPBA | String2: AMRRERCHAZBZA
Ekran Çıkısı:

```
Word1: AMRRERCHAZBZA
Word2: MAERBPHCAPPBA
Length of LCS: 7
All LCS:
AERCABA
MERCABA
AERHABA
MERHABA
```

Örnek 4:

String1: PESLIUITYFNIAZ | String2: EAWLINFMVNSAZ

Ekran Çıktısı:

```
----------------------
Word1: PESLIUITYFNIAZ
Word2: EAWLINFMVNSAZ
Length of LCS: 7
All LCS:
ELIFNAZ
```

Örnek 5:

String1: PKLOUMHLGN | String2: ASVZXFQWRE

Ekran Çıktısı:

```
----------------------------
Word1: PKLOUMHLGN
Word2: ASVZXFQWRE
Length of LCS: 0
!! No Common Subsequence !!
```