Gebze Technical University Department of Computer Engineering

CSE 654 / 484 Fall 2022

Homework 01

Report

Ömer Faruk Akduman

1801042094

Index

Homework Content

In this homework we will use edit distance to find similar text sections between documents with Smith-Waterman algorithm.

1. Download the standard textbooks from the Ministry of Education (http://aok.meb.gov.tr/kitap/) for at least 20 textbooks (literature, history, sociology, etc.) Convert them to text documents. Each document should be at most 400 lines. You may truncate the text if it is longer.

I downloaded 20 textbooks and converted to it text with truncate.

a1	2.11.2022 12:05	Microsoft Edge PD...	14.686 KB
a2	2.11.2022 12:06	Microsoft Edge PD...	23.037 KB
a3	2.11.2022 12:06	Microsoft Edge PD...	38.946 KB
a4	2.11.2022 11:24	Microsoft Edge PD...	16.568 KB
a5	5.11.2022 19:56	Microsoft Edge PD...	41.850 KB
a6	5.11.2022 19:20	Microsoft Edge PD...	14.686 KB
a7	5.11.2022 19:55	Microsoft Edge PD...	23.980 KB
a8	5.11.2022 19:55	Microsoft Edge PD...	22.070 KB
a9	5.11.2022 19:55	Microsoft Edge PD...	28.073 KB
a10	5.11.2022 19:55	Microsoft Edge PD...	16.568 KB
a11	5.11.2022 19:14	Microsoft Edge PD...	23.058 KB
a12	5.11.2022 19:14	Microsoft Edge PD...	15.011 KB
a13	5.11.2022 20:31	Microsoft Edge PD...	25.991 KB
a14	5.11.2022 20:30	Microsoft Edge PD...	20.311 KB
a15	5.11.2022 20:31	Microsoft Edge PD...	26.444 KB
a16	5.11.2022 20:10	Microsoft Edge PD...	22.960 KB
a17	5.11.2022 20:10	Microsoft Edge PD...	22.960 KB
a18	5.11.2022 20:31	Microsoft Edge PD...	32.067 KB
a19	5.11.2022 20:31	Microsoft Edge PD...	36.274 KB
a20	5.11.2022 20:10	Microsoft Edge PD...	44.961 KB

Also, I converted and truncate pdf files to text with the script code below. Added hw1.py in extra section

```
In [10]: import sys
         import codecs
         import PyPDF2
         from tqdm import tqdm


         count = 0
         def translateTotxt(file_name):
             global count
             # creating a pdf file object
             pdfFileObj = open(file_name  +".pdf", 'rb')
             file1 = open(file_name + ".txt","w")


             # creating a pdf reader object
             pdfReader = PyPDF2.PdfFileReader(pdfFileObj)

             num_pages = pdfReader.numPages
             print("num_pages")
             extracted_str = ""
             j = 0
             for i in range(int(num_pages)):
                 j+=1
                 if(j==5):
```

```
                 file1 = open(str(count) + ".txt","w")
                 try:
                     if(len(extracted_str)>2000):
                         extracted_str = extracted_str[:2000] #turnicate

                     print(extracted_str)
                     file1.write(extracted_str)
                 except:
                     print("there is an except")
                     count=-1
                 extracted_str = ""
                 file1.close()
                 count+=1
                 j=0
             extracted_str += pdfReader.getPage(i).extractText()
             print(len(extracted_str))
             file1.close()
             pdfFileObj.close()


         for i in tqdm(range(1, 3)):
             translateTotxt("a"+str(i))

100%|██████████████████████████████████████████| 2/2 [00:00<00:00,  5.27it/s]
```

Final txt files total 100 txt file (0-100)

0	6.11.2022 23:11	Metin Belgesi	3 KB
1	7.11.2022 02:44	Metin Belgesi	2 KB
2	7.11.2022 02:42	Metin Belgesi	2 KB
3	7.11.2022 02:42	Metin Belgesi	2 KB
4	7.11.2022 02:42	Metin Belgesi	2 KB
5	7.11.2022 02:42	Metin Belgesi	2 KB
6	7.11.2022 02:42	Metin Belgesi	2 KB
7	7.11.2022 02:42	Metin Belgesi	2 KB
8	7.11.2022 02:42	Metin Belgesi	2 KB
9	7.11.2022 02:42	Metin Belgesi	2 KB
10	7.11.2022 02:42	Metin Belgesi	2 KB
11	7.11.2022 02:42	Metin Belgesi	2 KB
12	7.11.2022 02:42	Metin Belgesi	2 KB
13	7.11.2022 02:42	Metin Belgesi	2 KB
14	7.11.2022 02:42	Metin Belgesi	2 KB
15	7.11.2022 02:42	Metin Belgesi	2 KB
16	7.11.2022 02:42	Metin Belgesi	2 KB
17	7.11.2022 03:36	Metin Belgesi	3 KB
18	7.11.2022 02:42	Metin Belgesi	2 KB
19	7.11.2022 02:42	Metin Belgesi	2 KB
20	7.11.2022 02:42	Metin Belgesi	2 KB

An example of txt files 0.txt

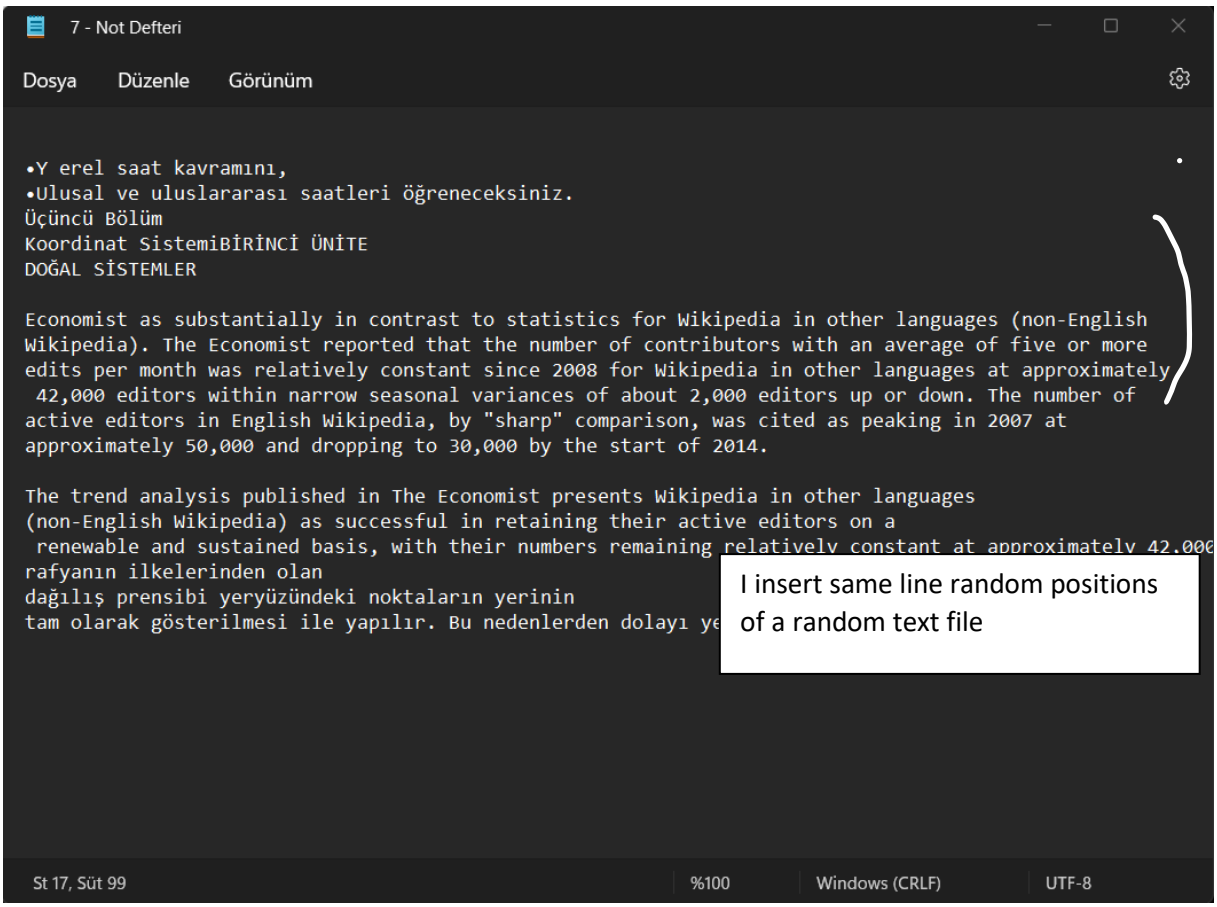
0 - Not Defteri

Dosya Düzenle Görünüm

```
•asking about and describing cities,
•identifying cultural differences,
•talking about travel and tourism,
•ordering foodFunctions and useful language11
  THEME  I BRIDGING CULTURES1.terminal
5.life jacket2.oxygen mask
6.passaport
9.baggage reclaim4. security check
8. boarding ticket1.Listen to the announcements and number the pictures.
(Anonsları dinleyiniz ve resimleri numaralandırınız.)LET'S LISTEN - L2.1
a)3.emergency exit
7.seat belt
10. flight attendant
c) d)
e) f)
g) h)
i) j)b)   THEME  I BRIDGING CULTURES12
2. Listen again and complete the sentences below.
(Tekrar dinleyin ve cümleleri tamamlayın.)
3. Look at the pictures and choose the ones related with travelling.
(Resimlere bakınız ve seyahat ile ilgili olanları işaretleyiniz.)1.Dear passengers, please go t
any baggage unattended.
2. Please, leave the plane in an orderly fashion, using the nearest _____ exit.3. Fligh
9.Please pass properly from the _____ points and prepare your papers in advance.
10. That's all passengers with express _____ and passengers travelling with young
children go to gate 3 for boarding. Thank you.
a)
```

St 1, Süt 1 %100 Windows (CRLF) UTF-8

2- Insert the same line of text into random positions of random text documents so that some of them have common text between them.



18 - Not Defteri

Dosya Düzenle Görünüm

```
160
Harita 1.33 Kutup İkliminin Dağılışı                                    Grafik 1. 21 Kutup İkl
O050100150200mm º C
Ş M M N H T A E E K A-50-45-40-30-25

Economist as substantially in contrast to statistics for Wikipedia in other languages (non-English
Wikipedia). The Economist reported that the number of contributors with an average of five or more
edits per month was relatively constant since 2008 for Wikipedia in other languages at approximately
 42,000 editors within narrow seasonal variances of about 2,000 editors up or down. The number of
active editors in English Wikipedia, by "sharp" comparison, was cited as peaking in 2007 at
approximately 50,000 and dropping to 30,000 by the start of 2014.
Kutup bölgelerinde mutlak nemin az olması havanın doyma noktasından
uzaklaşmasına neden olur bu durumun sonucu olarak kutup ikliminin yayıldı -
1.6.2 TÜRKİYE'DE İKLİM ÖZELLİKLERİ
1.6.2.1 Türkiye İklimine Etki Eden Faktörler
Bir ülkenin sahip olduğu iklim özellikleri, o ülkenin matematiksel konumu ve özel
konumu ile yakından ilgilidir. Türkiye' nin iklimine etki eden faktörleri de incelerken
iki başlık altında toplamak gerekir:
a. Matematiksel konuma göre iklimi etkileyen unsurlar
Türkiye 26º-45º Doğu boylamları 36º-42º kuzey enlemleri arasında yer alır. Bulun -
duğu enlem değerlerine göre orta kuşak ülkesidir ve ılıman iklim kuşağında yer alır.
Dünya iklim sınıflandırmasında ise Akdeniz iklim kuşağında bulunur. Bu durumun
sonuçlarına bakıldığında şöyle bir sıralama yapabiliriz.
Şekil
```

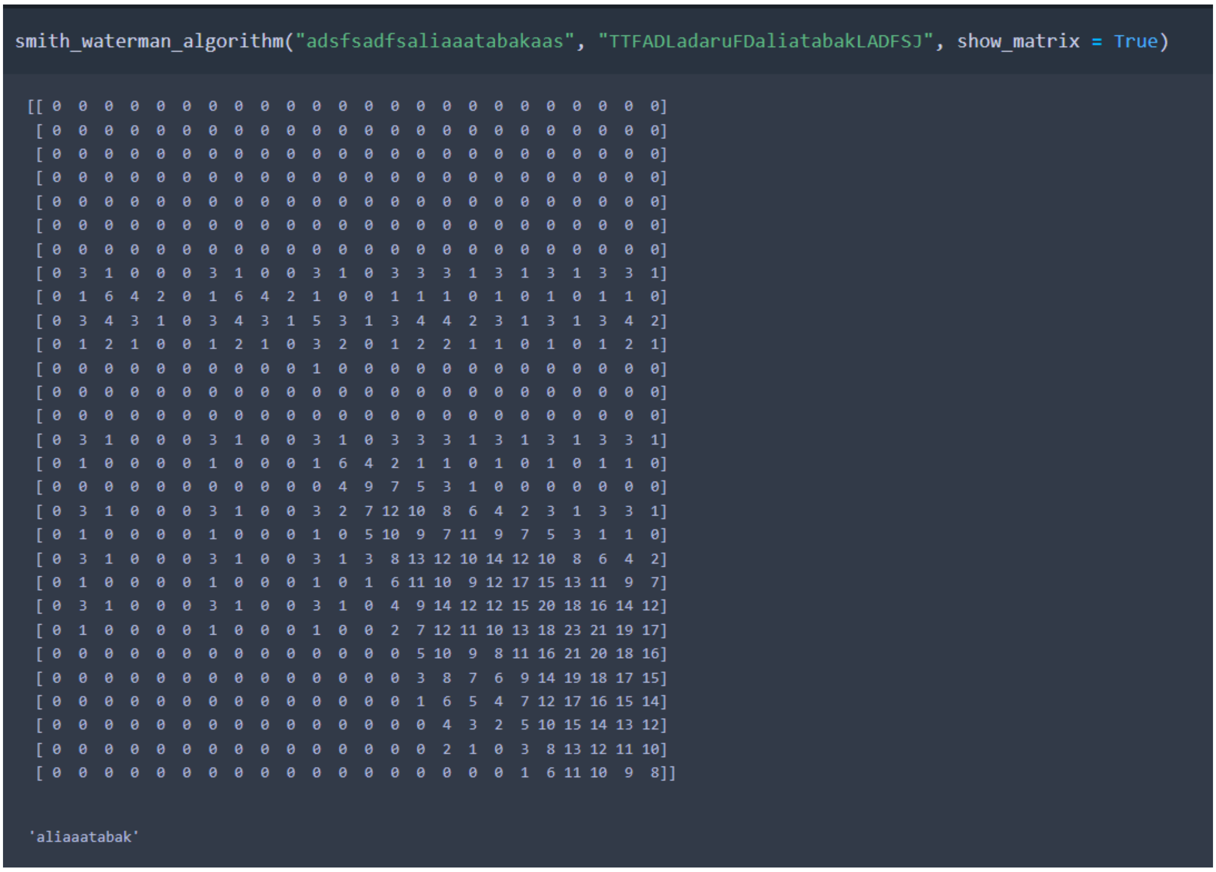
I insert same line random positions of a random text file

St 14, Süt 1 %100 Windows (CRLF) UTF-8

3-Using Smith-Waterman Algorithm, find the common lines between given two texts.

I implement Smith-Waterman Algorithm and found the common lines between given two texts.

Here is an example that given 2 strings.

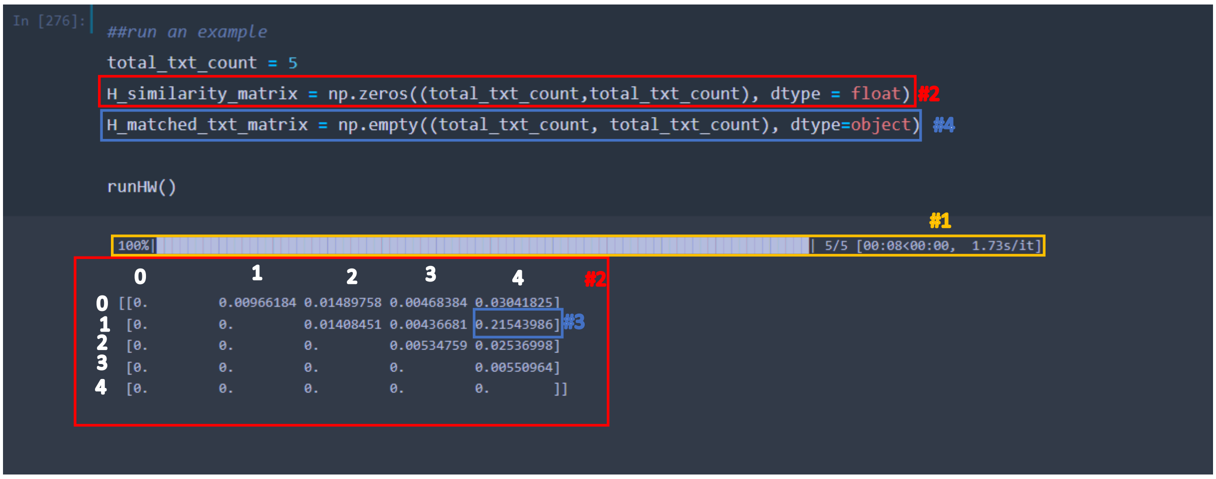


```
smith_waterman_algorithm("adsfsadfsaliaaatabakaas", "TTFADLadaruFDaliatabakLADFSJ", show_matrix = True)
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  3  1  0  0  3  1  0  3  3  3  1  3  1  3  1  3  3  1]
 [ 0  1  6  4  2  0  1  6  4  2  1  0  0  1  1  1  0  1  0  1  0  1  1  0]
 [ 0  3  4  3  1  0  3  4  3  1  5  3  1  3  4  4  2  3  1  3  1  3  4  2]
 [ 0  1  2  1  0  0  1  2  1  0  3  2  0  1  2  2  1  1  0  1  0  1  2  1]
 [ 0  0  0  0  0  0  0  0  0  0  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  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  3  1  0  0  3  1  0  3  3  3  1  3  1  3  1  3  3  1]
 [ 0  1  0  0  0  0  1  0  0  0  1  6  4  2  1  1  0  1  0  1  0  1  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  4  9  7  5  3  1  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  3  1  0  0  3  2  7 12 10  8  6  4  2  3  1  3  3  1]
 [ 0  1  0  0  0  0  1  0  0  0  1  0  5 10  9  7 11  9  7  5  3  1  1  0]
 [ 0  3  1  0  0  0  3  1  0  0  3  1  3  8 13 12 10 14 12 10  8  6  4  2]
 [ 0  1  0  0  0  0  1  0  0  0  1  0  1  6 11 10  9 12 17 15 13 11  9  7]
 [ 0  3  1  0  0  0  3  1  0  0  3  1  0  4  9 14 12 12 15 20 18 16 14 12]
 [ 0  1  0  0  0  0  1  0  0  0  1  0  0  2  7 12 11 10 13 18 23 21 19 17]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  5 10  9  8 11 16 21 20 18 16]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  3  8  7  6  9 14 19 18 17 15]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  6  5  4  7 12 17 16 15 14]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  4  3  2  5 10 15 14 13 12]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  1  0  3  8 13 12 11 10]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  6 11 10  9  8]]


'aliaaatabak'
```

Here is an example about given 5 txt file



```
In [276]:
##run an example
total_txt_count = 5
H_similarity_matrix = np.zeros((total_txt_count,total_txt_count), dtype = float)
H_matched_txt_matrix = np.empty((total_txt_count, total_txt_count), dtype=object)

runHW()
```

```
100%|██████████████████████████████████████████████████| 5/5 [00:08<00:00,  1.73s/it]
[[0.         0.00966184 0.01489758 0.00468384 0.03041825]
 [0.         0.         0.01408451 0.00436681 0.21543986]
 [0.         0.         0.         0.00534759 0.02536998]
 [0.         0.         0.         0.         0.00550964]
 [0.         0.         0.         0.         0.        ]]
```

#1 Progress bar that represents the timing of process

#2 Similarity matrix that represents text similarity files each other for example #3 represents “the blue one” similarity of the file 1.txt and file 4.txt

#3 similarity rate calculates with the *longest-matched-word/total-word*

#4 a container matrix containing the matched-longest word

```
print(H_matched_txt_matrix[1][4])
print(H_matched_txt_matrix[0][1])

 _          _       _ _ _ _ _ _ _ _ _ _      _
about and describing
```

Common lines with given files

4. Define the cost of substitution, deletion, insertion yourself.

- Substitution matrix: match = 3, mismatch=-3
- Gap penalty: -2 (a linear gap penalty of)

from wikipedia

5- Some examples of my implementation of Waterman-Smith Algorithm

Example for 20 text file

Recall: file 7.txt and 18.txt are inserted same text line part

```
In [319]:
total_txt_count = 20
H_similarity_matrix = np.zeros((total_txt_count,total_txt_count), dtype = float)
H_matched_txt_matrix = np.empty((total_txt_count, total_txt_count), dtype=object)

runHW()

100%|| 20/20 [01:38<00:00,  4.90s/it]
```

```
In [322]:
#find most similar files pair
i,j = np.unravel_index(H_similarity_matrix.argmax(), H_similarity_matrix.shape)

print(str(i)+".txt and "+str(j)+".txt" + " are have most similar text")

print("Matched text is: " +  H_matched_txt_matrix[i][j] +
      " with " + str(H_similarity_matrix[i][j]) + " score")

7.txt and 18.txt are have most similar text
Matched text is: as substantially in contrast to statistics for Wikipedia in other languages (non-English
Wikipedia). The Economist reported that the number of contributors with an average of five or more
edits per month was relatively constant since 2008 for Wikipedia in other languages at approximately
 42,000 editors within narrow seasonal variances of about 2,000 editors up or down. The number of
active editors in English Wikipedia, by "sharp" comparison, was cited as peaking in 2007 at
approximately 50,000 and dropping to 30,000 by the start of with 0.4148148148148148 score
```

Score matrix example

```
In [323]: smith_waterman_algorithm("G C C T C N T C C C G G C", "G G G C N T C C G G G F H", show_matrix = True)
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  0  0  0  0  0  3  3  1]
 [ 0  3  1  0  0  0  0  0  0  0  0  3  6  4]
 [ 0  3  1  0  0  0  0  0  0  0  0  3  6  4]
 [ 0  1  6  4  2  3  1  0  3  3  3  1  4  9]
 [ 0  0  4  3  1  1  6  4  2  1  1  0  2  7]
 [ 0  0  2  1  6  4  4  9  7  5  3  1  0  5]
 [ 0  0  3  5  4  9  7  7 12 10  8  6  4  3]
 [ 0  0  3  6  4  7  6  5 10 15 13 11  9  7]
 [ 0  3  1  4  3  5  4  3  8 13 12 16 14 12]
 [ 0  3  1  2  1  3  2  1  6 11 10 15 19 17]
 [ 0  3  1  0  0  1  0  0  4  9  8 13 18 16]
 [ 0  1  0  0  0  0  0  0  2  7  6 11 16 15]
 [ 0  0  0  0  0  0  0  0  0  5  4  9 14 13]]


'G G G C C _ C N T C C C G G'
```

```
In [324]: smith_waterman_algorithm("A B C D", "D D C D", show_matrix = True)
```

```
[[0 0 0 0 0]
 [0 0 0 0 3]
 [0 0 0 0 3]
 [0 0 0 3 1]
 [0 0 0 1 6]]


'C D'
```

```
In [325]: smith_waterman_algorithm("f k j d a l f a ; d s k f j ; a f", "f a d l s k j ; f a s d j l ; k f s d a l f j
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  0  3  1  0  0  0  0  3  1  0  0  3]
 [ 0  1  0  0  0  3  1  1  6  4  2  0  0  1  0  0  3  1]
 [ 0  0  0  0  3  1  0  0  4  3  7  5  3  1  0  0  1  0]
 [ 0  0  0  0  1  0  4  2  2  1  5  4  2  0  0  0  0  0]
 [ 0  0  0  0  0  0  2  1  0  0  3  8  6  4  2  0  0  0]
 [ 0  0  3  1  0  0  0  0  0  0  1  6 11  9  7  5  3  1]
 [ 0  0  1  6  4  2  0  0  0  0  0  4  9  8 12 10  8  6]
 [ 0  0  0  4  3  1  0  0  0  3  1  2  7  6 10 15 13 11]
 [ 0  3  1  2  1  0  0  3  1  1  0  0  5 10  8 13 12 16]
 [ 0  1  0  0  0  4  2  1  6  4  2  0  3  8  7 11 16 14]
 [ 0  0  0  0  0  2  1  0  4  3  1  5  3  6  5  9 14 13]
 [ 0  0  0  0  3  1  0  0  2  1  6  4  2  4  3  7 12 11]
 [ 0  0  0  3  1  0  0  0  0  0  4  3  1  2  7  5 10  9]
 [ 0  0  0  1  0  0  3  1  0  0  2  1  0  0  5  4  8  7]
 [ 0  0  0  0  0  0  1  0  0  3  1  0  0  0  3  8  6  5]
 [ 0  0  3  1  0  0  0  0  0  1  0  0  3  1  1  6  5  3]
 [ 0  3  1  0  0  0  0  3  1  0  0  0  1  6  4  4  3  8]
 [ 0  1  0  0  0  0  0  1  0  0  0  3  1  4  3  2  1  6]
 [ 0  0  0  0  3  1  0  0  0  0  3  1  0  2  1  0  0  4]
 [ 0  0  0  0  1  6  4  2  3  1  1  0  0  0  0  0  3  2]
 [ 0  0  0  0  0  4  9  7  5  3  1  0  0  0  0  0  1  0]
 [ 0  3  1  0  0  2  7 12 10  8  6  4  2  3  1  0  0  4]
 [ 0  1  0  4  2  0  5 10  9  7  5  3  1  1  6  4  2  2]
 [ 0  0  0  2  1  0  3  8  7  6  4  2  0  0  4  3  1  0]]


'f a _ d _ s k _ j ; _ f'
```

Score matrix examples with recursive functions:

```
smith_waterman_algorithmR("yine yaz kalemim ile", "kale yaziyorum yine ve", show_matrix = True)
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  3  1  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  3  1  0  1  6  4  2  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  1  0  0  0  4  9  7  5  3  1  0  0  3  1]
 [ 0  0  0  0  3  1  0  0  0  0  0  2  7 12 10  8  6  4  2  1  6]
 [ 0  0  0  0  1  6  4  2  0  3  1  0  5 10  9  7  5  9  7  5  4]
 [ 0  3  1  0  0  4  9  7  5  3  1  0  3  8  7  6  4  7  6  4  2]
 [ 0  1  0  0  0  2  7 12 10  8  6  4  2  6  5  4  3  5  4  3  1]
 [ 0  0  0  0  0  0  5 10 15 13 11  9  7  5  3  2  1  3  2  1  0]
 [ 0  0  3  1  0  0  3  8 13 12 10  8  6  4  2  6  4  2  6  4  2]
 [ 0  3  1  0  0  0  3  6 11 10  9  7  5  3  1  4  3  1  4  3  1]
 [ 0  1  0  0  0  0  1  4  9  8  7  6  4  2  0  2  1  0  2  1  0]
 [ 0  0  0  0  0  0  0  2  7  6  5  4  3  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  5  4  3  2  1  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  3  2  1  0  0  0  3  1  3  1  0  0  0]
 [ 0  0  0  0  0  3  1  0  1  6  4  2  0  0  1  0  1  6  4  2  0]
 [ 0  3  1  0  0  1  6  4  2  4  3  1  0  0  0  0  0  4  3  1  0]
 [ 0  1  6  4  2  0  4  3  1  2  1  0  0  0  0  3  1  2  7  5  3]
 [ 0  0  4  9  7  5  3  1  0  0  0  0  0  0  0  1  0  0  5  4  2]
 [ 0  0  2  7 12 10  8  6  4  2  0  0  0  3  1  0  0  0  3  2  7]
 [ 0  0  0  5 10 15 13 11  9  7  5  3  1  1  0  0  0  3  1  0  5]
 [ 0  0  0  3  8 13 12 10  8  6  4  2  0  0  0  0  0  1  0  0  3]
 [ 0  0  0  1  6 11 10  9  7  5  3  1  0  3  1  0  0  0  0  0  3]]


'e yaz'
```

```
In [336]: smith_waterman_algorithmR("hala nieye", "halama bayrama gidiyorum", show_matrix = True)
```

```
[[ 0  0  0  0  0  0  0  0  0  0  0]
 [ 0  3  1  0  0  0  0  0  0  0  0]
 [ 0  1  6  4  3  1  0  0  0  0  0]
 [ 0  0  4  9  7  5  3  1  0  0  0]
 [ 0  0  3  7 12 10  8  6  4  2  0]
 [ 0  0  1  5 10  9  7  5  3  1  0]
 [ 0  0  3  3  8  7  6  4  2  0  0]
 [ 0  0  1  1  6 11  9  7  5  3  1]
 [ 0  0  0  0  4  9  8  6  4  2  0]
 [ 0  0  3  1  3  7  6  5  3  1  0]
 [ 0  0  1  0  1  5  4  3  2  6  4]
 [ 0  0  0  0  0  3  2  1  0  4  3]
 [ 0  0  3  1  3  1  0  0  0  2  1]
 [ 0  0  1  0  1  0  0  0  0  0  0]
 [ 0  0  3  1  3  1  0  0  0  0  0]
 [ 0  0  1  0  1  6  4  2  0  0  0]
 [ 0  0  0  0  0  4  3  1  0  0  0]
 [ 0  0  0  0  0  2  1  6  4  2  0]
 [ 0  0  0  0  0  0  0  4  3  1  0]
 [ 0  0  0  0  0  0  0  3  1  0  0]
 [ 0  0  0  0  0  0  0  1  0  4  2]
 [ 0  0  0  0  0  0  0  0  0  2  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]]

'hala'
```

Bibliography

Wikipedia