INFORMATION RETRIEVAL

PRACTICAL-2

NAME: Neeraj Appari T073

Aim: Implement Dynamic programming algorithm for computing the edit distance between strings s1 and s2.

DESCRIPTION:

Levenshtein distance algorithm

The Levenshtein algorithm calculates the least number of edit

operations that are necessary to modify one string to obtain another

string

Steps:

1. A matrix is initialized measuring in the (m, n) cell the Levenshtein distance between the m-character prefix of one with the n-prefix of the other word.

2. The matrix can be filled from the upper left to the lower right corner.

3. Each jump horizontally or vertically corresponds to an insert or a delete, respectively.

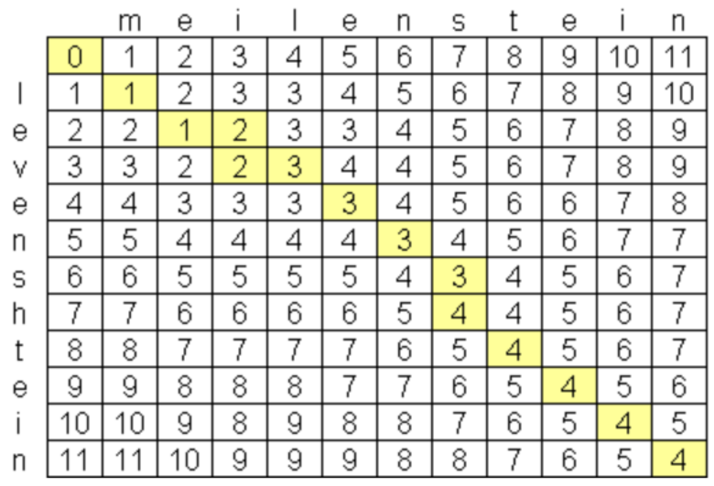
4. The cost is normally set to 1 for each of the operations.

5. The diagonal jump can cost either one, if the two characters in the row and column do not match else 0, if they match. Each cell always minimizes the cost locally.

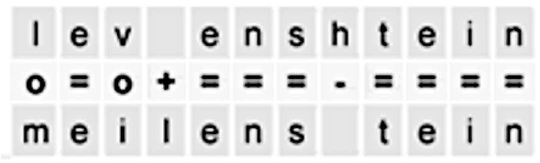
6. This way the number in the lower right corner is the Levenshtein distance between both words.

Levenshtein Distance Algorithm
Example:

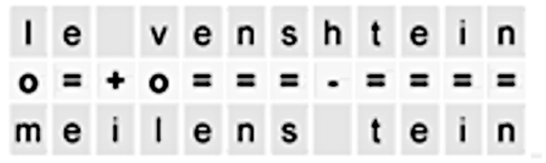
```
EDITDISTANCE($s_1$, $s_2$)
int $m[i,j] = 0$
for $i \leftarrow 1$ to $|s_1|$
do $m[i,0] = i$
for $j \leftarrow 1$ to $|s_2|$
do $m[0,j] = j$
for $i \leftarrow 1$ to $|s_1|$
do for $j \leftarrow 1$ to $|s_2|$
    do $m[i,j] = \min\{m[i-1,j-1] + \text{if } (s_1[i] = s_2[j]) \text{ then } 0 \text{ else } 1\text{fi},$
                      $m[i-1,j] + 1,$
                      $m[i,j-1] + 1\}$
return $m[|s_1|, |s_2|]$
```



		m	e	i	l	e	n	s	t	e	i	n
	0	1	2	3	4	5	6	7	8	9	10	11
l	1	1	2	3	3	4	5	6	7	8	9	10
e	2	2	1	2	3	3	4	5	6	7	8	9
v	3	3	2	2	3	4	4	5	6	7	8	9
e	4	4	3	3	3	3	4	5	6	6	7	8
n	5	5	4	4	4	4	3	4	5	6	7	7
s	6	6	5	5	5	5	4	3	4	5	6	7
h	7	7	6	6	6	6	5	4	4	5	6	7
t	8	8	7	7	7	7	6	5	4	5	6	7
e	9	9	8	8	8	7	7	6	5	4	5	6
i	10	10	9	8	9	8	8	7	6	5	4	5
n	11	11	10	9	9	9	8	8	7	6	5	4



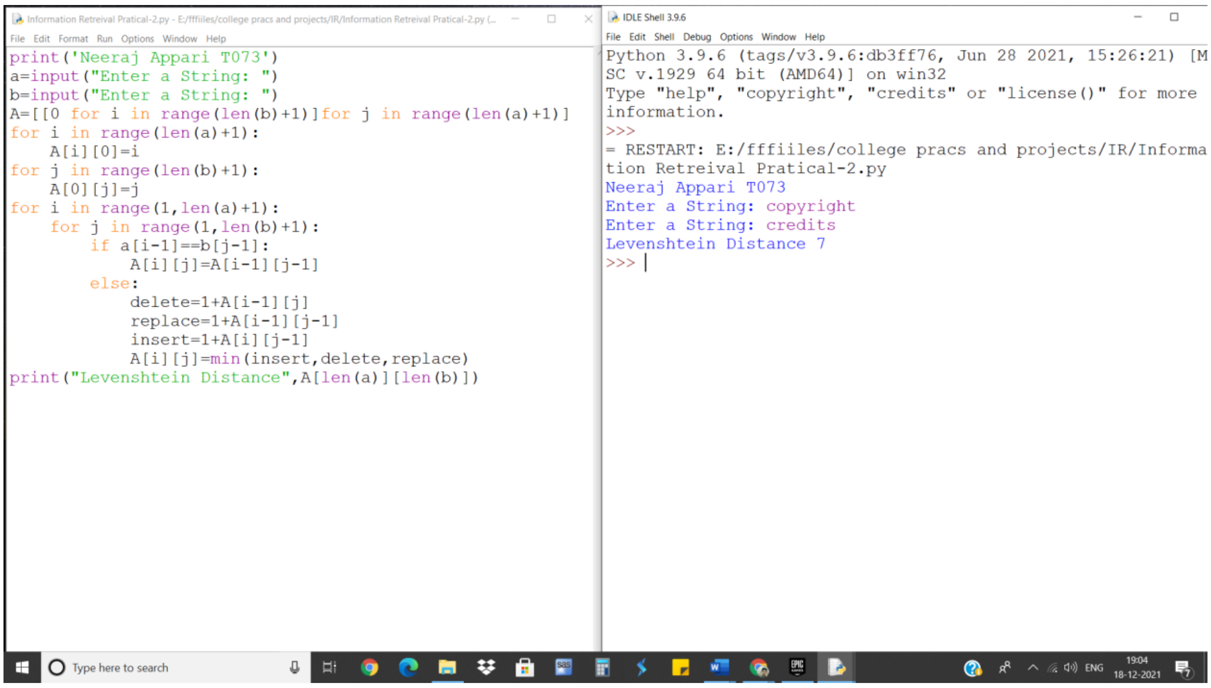
l	e	v		e	n	s	h	t	e	i	n
o	=	o	+	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n



l	e		v	e	n	s	h	t	e	i	n
o	=	+	o	=	=	=	-	=	=	=	=
m	e	i	l	e	n	s		t	e	i	n

```
print('Name and Roll No')
a=input("Enter a String: ")
b=input("Enter a String: ")
A=[[0 for i in range(len(b)+1)]for j in range(len(a)+1)]
for i in range(len(a)+1):
    A[i][0]=i
for j in range(len(b)+1):
    A[0][j]=j
for i in range(1,len(a)+1):
    for j in range(1,len(b)+1):
        if a[i-1]==b[j-1]:
            A[i][j]=A[i-1][j-1]
        else:
            delete=1+A[i-1][j]
            replace=1+A[i-1][j-1]
            insert=1+A[i][j-1]
            A[i][j]=min(insert,delete,replace)
print("Levenshtein Distance",A[len(a)][len(b)])
```

Screenshot:



```
print('Neeraj Appari T073')
a=input("Enter a String: ")
b=input("Enter a String: ")
A=[[0 for i in range(len(b)+1)]for j in range(len(a)+1)]
for i in range(len(a)+1):
    A[i][0]=i
for j in range(len(b)+1):
    A[0][j]=j
for i in range(1,len(a)+1):
    for j in range(1,len(b)+1):
        if a[i-1]==b[j-1]:
            A[i][j]=A[i-1][j-1]
        else:
            delete=1+A[i-1][j]
            replace=1+A[i-1][j-1]
            insert=1+A[i][j-1]
            A[i][j]=min(insert,delete,replace)
print("Levenshtein Distance",A[len(a)][len(b)])
```

```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [M
SC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
information.
>>>
= RESTART: E:/fffiiles/college pracs and projects/IR/Informa
tion Retreival Pratical-2.py
Neeraj Appari T073
Enter a String: copyright
Enter a String: credits
Levenshtein Distance 7
>>>
```

Information Retreival Pratical-2.py - E:/fffiiles/college pracs and projects/IR/Information Retreival Pratical-2.py (...
File Edit Format Run Options Window Help
IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Type here to search
ENG
19:04
18-12-2021