Mahir Efe KAYA
2016400195

CMPE 487

Assignment 1

Mahir Efe KAYA-2016400195

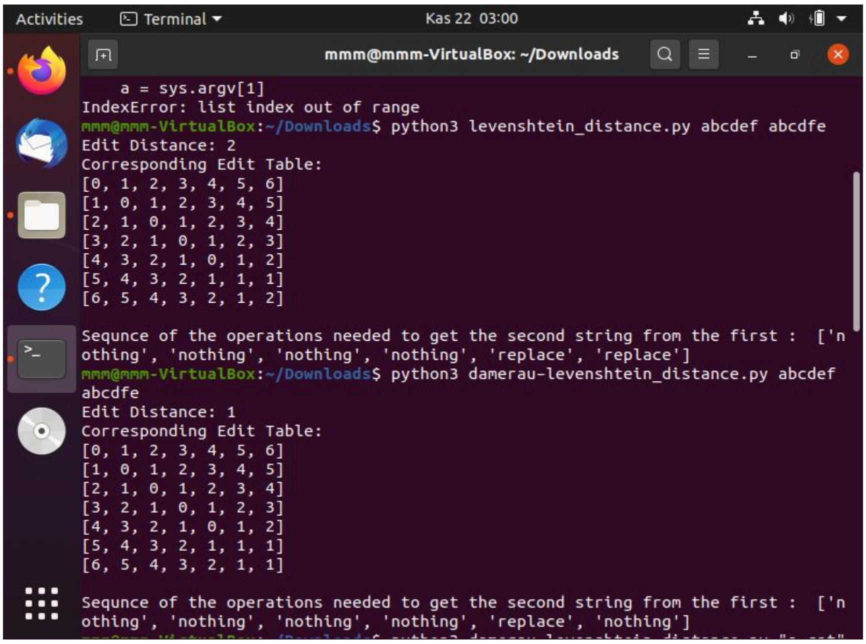
The main algorithm for filling the table:

- Checks if the same character exists in the comparison, if so replacecost + 0 else replacecost +1
- Takes the minimum value of the deletecost and insertion cost of that square unit with the replacecost, and writes the result as the minimum value for that square unit.
- Edges are in increasing order starting from the left top corner.
- Left top corner is 0

1) **In the first question**, I created a table to hold all the change values, to later backtrack from its right bottom corner. Backtracking would be needed for to get the sequence of moves needed to change the string a to string b.
   To backtrack, I had started from one corner to the opposite corner, changing the coordinates accordingly to the minimum values around the current unit square. The directions I take decides the sequence of the moves.

2) **In the second question**, other than adding a check for swapping, nothing new has been added to the main algorithm for filling the table. However I created another table to later backtrack from it. It stores the directions the algorithm taken, to later follow through it. Since it would be much more complex with the contribution of the last check for swapping, I thought using a second table to store the directions would be much more helpful and easier to understand. With the direction check, I also used a check for the difference since there could be no moves done for that unit square in the cases of the same characters.

Appendices

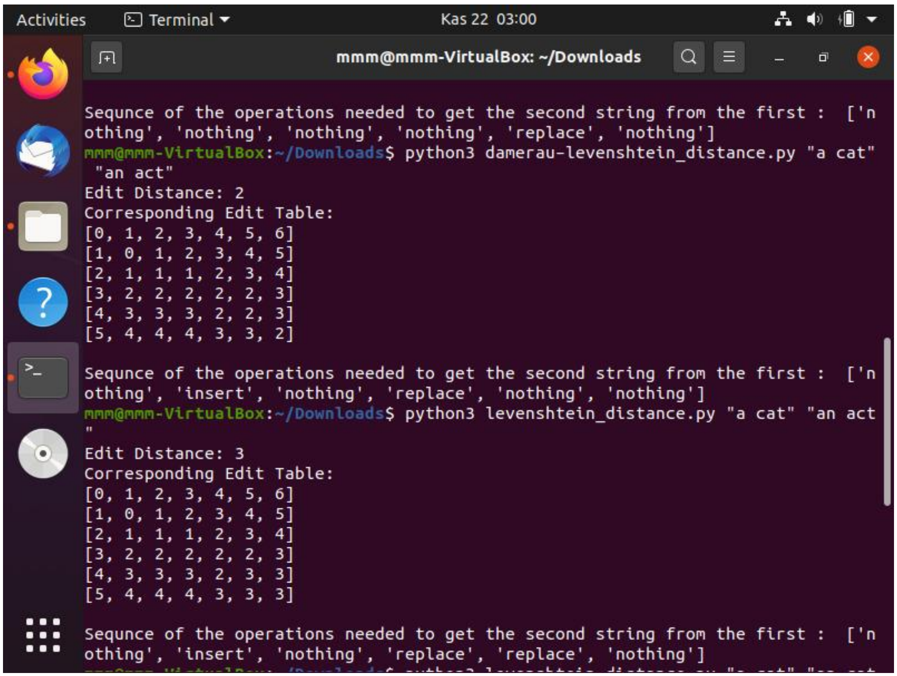


```
    a = sys.argv[1]
IndexError: list index out of range
mmm@mmm-VirtualBox:~/Downloads$ python3 levenshtein_distance.py abcdef abcdfe
Edit Distance: 2
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6]
[1, 0, 1, 2, 3, 4, 5]
[2, 1, 0, 1, 2, 3, 4]
[3, 2, 1, 0, 1, 2, 3]
[4, 3, 2, 1, 0, 1, 2]
[5, 4, 3, 2, 1, 1, 1]
[6, 5, 4, 3, 2, 1, 2]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'nothing', 'nothing', 'nothing', 'replace', 'replace']
mmm@mmm-VirtualBox:~/Downloads$ python3 damerau-levenshtein_distance.py abcdef
abcdfe
Edit Distance: 1
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6]
[1, 0, 1, 2, 3, 4, 5]
[2, 1, 0, 1, 2, 3, 4]
[3, 2, 1, 0, 1, 2, 3]
[4, 3, 2, 1, 0, 1, 2]
[5, 4, 3, 2, 1, 1, 1]
[6, 5, 4, 3, 2, 1, 1]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'nothing', 'nothing', 'nothing', 'replace', 'nothing']
```

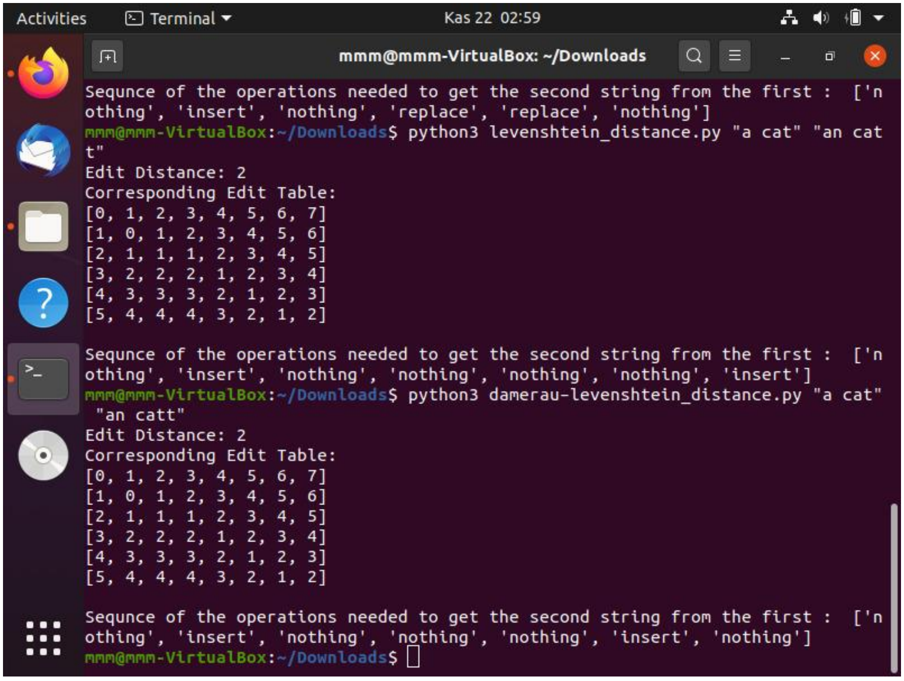
Mahir Efe KAYA
2016400195



```
Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'nothing', 'nothing', 'nothing', 'replace', 'nothing']
mmm@mmm-VirtualBox:~/Downloads$ python3 damerau-levenshtein_distance.py "a cat"
 "an act"
Edit Distance: 2
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6]
[1, 0, 1, 2, 3, 4, 5]
[2, 1, 1, 1, 2, 3, 4]
[3, 2, 2, 2, 2, 2, 3]
[4, 3, 3, 3, 2, 2, 3]
[5, 4, 4, 4, 3, 3, 2]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'insert', 'nothing', 'replace', 'nothing', 'nothing']
mmm@mmm-VirtualBox:~/Downloads$ python3 levenshtein_distance.py "a cat" "an act
"
Edit Distance: 3
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6]
[1, 0, 1, 2, 3, 4, 5]
[2, 1, 1, 1, 2, 3, 4]
[3, 2, 2, 2, 2, 2, 3]
[4, 3, 3, 3, 2, 3, 3]
[5, 4, 4, 4, 3, 3, 3]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'insert', 'nothing', 'replace', 'replace', 'nothing']
```



```
Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'insert', 'nothing', 'replace', 'replace', 'nothing']
mmm@mmm-VirtualBox:~/Downloads$ python3 levenshtein_distance.py "a cat" "an cat
t"
Edit Distance: 2
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6, 7]
[1, 0, 1, 2, 3, 4, 5, 6]
[2, 1, 1, 1, 2, 3, 4, 5]
[3, 2, 2, 2, 1, 2, 3, 4]
[4, 3, 3, 3, 2, 1, 2, 3]
[5, 4, 4, 4, 3, 2, 1, 2]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'insert', 'nothing', 'nothing', 'nothing', 'nothing', 'insert']
mmm@mmm-VirtualBox:~/Downloads$ python3 damerau-levenshtein_distance.py "a cat"
 "an catt"
Edit Distance: 2
Corresponding Edit Table:
[0, 1, 2, 3, 4, 5, 6, 7]
[1, 0, 1, 2, 3, 4, 5, 6]
[2, 1, 1, 1, 2, 3, 4, 5]
[3, 2, 2, 2, 1, 2, 3, 4]
[4, 3, 3, 3, 2, 1, 2, 3]
[5, 4, 4, 4, 3, 2, 1, 2]

Sequnce of the operations needed to get the second string from the first :  ['n
othing', 'insert', 'nothing', 'nothing', 'nothing', 'insert', 'nothing']
mmm@mmm-VirtualBox:~/Downloads$ 
```