Instructions

- Solve the problem on the following page in python.
- You should not need to use any external library.
- There is no time limit or necessary right answer. (The problem can in fact be solved in 2 ways: recursively or using dynamic programming). Attempt to do it in both ways if time permits. If not, pick one way of doing it but have a specific reason in mind for your choice.
- We are looking at the quality of your solution and the programmatic style. Try to make it as easy as possible to read your code - feel free to add comments, etc.
- What we are interested in seeing is the way you think. Send your solution by email to nisarg@vector.ai in a file called ***edit_distance.py*** with the subject line "Software Engineering Test". We will subsequently schedule a 30 minute session to talk through your solution.
- Most importantly, enjoy it!

Problem

Given two strings str1 and str2 with the operations below that can performed on str1, find the minimum number of edits (operations) required to convert str1 into str2.

1. Insert
2. Remove
3. Replace

This is otherwise known as the Levenshtein distance between 2 strings.

Examples

Input: str1 = "geek", str2 = "gesek"
Output: 1
We can convert str1 into str2 by inserting a 's'.

Input: str1 = "cat", str2 = "cut"
Output: 1
We can convert str1 into str2 by replacing 'a' with 'u'.

Input: str1 = "sunday", str2 = "saturday"
Output: 3
Last three and first characters are same. We basically
need to convert "un" to "atur". This can be done using
below three operations.
Replace 'n' with 'r', insert t, insert a