

# Assignment: Python String Edit distance

When working with real world text you will often come across slight misspellings. For example, a person looking for the University of Oklahoma will often misspell the name of the state as “Oaklahoma”. As responsible Sooners, we have to ensure that no such mistakes exist.

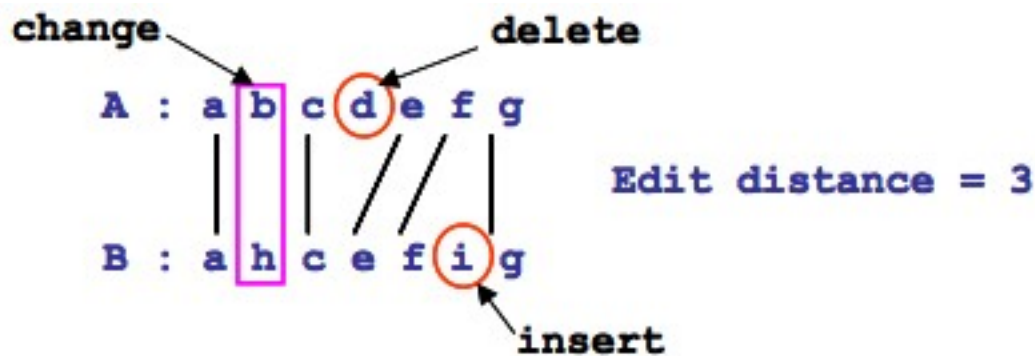
In this activity, we will delve deeper into the usage of Python by implementing one popular algorithm for spelling and error correction. We will code the string “edit distance” algorithm.

## String Edit Distance

Given two strings  $s_1$  and  $s_2$  that contain lowercase ASCII or Unicode characters. The edit distance between  $s_1$  and  $s_2$  is the minimum number of edit operations needed to convert  $s_1$  into  $s_2$ . There are three possible edit operations:

- (i) **change**: replace one character of  $s_1$  by another single character of  $s_2$ .
- (ii) **delete**: delete one character from  $s_1$ .
- (iii) **insert**: insert one character of  $s_2$  into  $s_1$

In the example below, we execute the following function `distance('abcdefg', 'ahcefig')`. The edit distance is three and includes a change, delete, and insert operation. The edit operations are a change (i.e., replacing b of A by h of B), a deletion (i.e., deleting d from A), and an insertion (i.e., inserting i of B into A). Many possible combinations of edits may be executed to transform one string to another — but we would like the minimum number of edits. This is also called edit distance. In this example, and in this assignment, each edit operation should have a cost of one unit.



## Submission Instructions

Create a single python file called `edit_distance.py`. Within the file create a function called `distance` that takes at least two parameters (strings). Each input string is between 2 and 10 characters in length. This function should return the minimum number of edits to transform one parameter to the other. Use the above string edit distance algorithm to accomplish this. It is not necessary to use the dynamic programming solution to complete this.

## Grading Criteria

This assignment is here to help you get acquainted with the Python programming language and also solve a common problem prevalent in text. Plagiarism will be treated as a violation of academic integrity.

Program runs: 70%

Produces correct result: 30%

The program will be run by a script similar to the one below.

```
$ python
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import edit_distance
```

```
>> from edit_distance import distance
```

```
>>> distance("test1a","test1b") == 1
```