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 is a technique called string edit distance.

# String Edit Distance

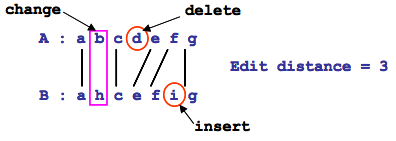
Given two strings s1 and s2 that contain lowercase ascii characters. The edit distance between s1 and s2 is the minimum number of edit operations needed to convert s1 into s2. The three edit operations are the following:

(i) change: replace one character of s1 by another single character of s2.

(ii) deletion: delete one character from s1.

(iii) insertion: insert one character of s2 into s1

In the example below we execute the following function *distance*(‘abcdefg’, ahcefig’). The edit distance is three and incudes 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 bay be executed to transform one string to another — but we would like the minimum number of edits. This is also called Levenshtein distance.



# Submission Instructions

Create a 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 1 and 50 characters in length. If you are using Python 2, you can safely assume that each input string is a Unicode object. 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

>>> edit\_distance.distance(test1a,testb) == test1c