In [2]:

```
import numpy as np
import pandas as pd
```

In [ ]:

```python
# Function to find Levenshtein distance between string `X` and `Y`.
def edit_dist(X, Y):
    # `m` and `n` is the total number of characters in `X` and `Y`, respectively
    (m, n) = (len(X), len(Y))

    # For all pairs of `i` and `j`, `T[i, j]` will hold the Levenshtein distance
    # between the first `i` characters of `X` and the first `j` characters of `Y`.
    # Note that `T` holds `(m+1)×(n+1)` values.
    T = [[0 for x in range(n + 1)] for y in range(m + 1)]
    print("-"*50)
    print(np.array(T))
    print("-"*50)

    # we can transform source prefixes into an empty string by
    # dropping all characters
    for i in range(1, m + 1):
        T[i][0] = i
        print(np.array(T))
        print("-"*50)

    # we can reach target prefixes from empty source prefix
    # by inserting every character
    for j in range(1, n + 1):
        T[0][j] = j
        print(np.array(T))
        print("-"*50)

    # fill the lookup table in a bottom-up manner
    for i in range(1, m + 1):

        for j in range(1, n + 1):
            if X[i - 1] == Y[j - 1]:
                cost = 0
            else:
                cost = 2

            T[i][j] = min(T[i - 1][j] + 1,      # deletion
                          T[i][j - 1] + 1,      # insertion
                          T[i - 1][j - 1] + cost) # replace

    print(np.array(T))
```

```
    return T[m][n]



if __name__ == '__main__':

    X = 'kitten'    # 6
    Y = 'sitting'   # 7

    print('The Levenshtein distance is', edit_dist(X, Y))
```