

# Final Report

Student Name: Balkar Singh

## Introduction

The purpose of this report is to analyze the Levenshtein distance algorithm and its performance on different string. The edit minimum distance calculated the from insertions, deletions, or substitutions required to change one string into another.

## Levenshtein algorithm implementation

- Import time module
- User input for string 1 and string 2
- Create a 2D array to from 2D zero matrix
- Compute edit distance (also know as levenshtein distance) using function levenshtein\_algorithm
- Printing

Display the edit\_distance.

Display the matrix

Measure and display execution time.

### Test 1 :

```
Enter the First String: fat
Enter the Second String: cat
From fat to cat

Matrix Generated
0  1  2  3
1  1  2  3
2  2  1  2
3  3  2  1

Levenshtein Distance: 1

Execution time: 0.003839 seconds
```

### Test 2 :

```
Enter the First String: silent
Enter the Second String: listen

Matrix Generated
0  1  2  3  4  5  6
1  1  2  2  3  4  5
2  2  1  2  3  4  5
3  2  2  2  3  4  5
4  3  3  3  3  3  4
5  4  4  4  4  4  3
6  5  5  5  4  5  4

Levenshtein Distance: 4

Execution time: 0.004010 seconds
```

## Observation

From the above two test results the time recorded moving From fat to cat **is 0.003839** seconds with Levenshtein Distance: **1** and From silent to listen **is 0.004010** seconds with Levenshtein Distance: **4**. The execution time (excluding user input delay) is calculation by the Python time library using `time.perf_counter()`

The time and space complexity depends upon the first string and second string for this program which is represent as  $O(m * n)$ .

## Application Program – Spelling Suggestion

### Implementation

- Import the json module for working with the dictionary file.
- Import the time module to measure execution time.
- Import the levenshtein\_algorithm function from levenshtein\_algo.py for computing edit distances.
- Load Dictionary:

Reading dict2.json as file.

Loading JSON dictionary to extract word list.

- Define function `spelling_suggestions(input_string, dictionary)`

Compute Levenshtein distance for each word in the dictionary.

Find the minimum distance and its matching words.

Return the matching word(s) and their distance(s).

- Main Execution:

Input a String/Sentence from the user.

Split the sentence into words and store as list.

Step to start timer to avoid user input delay - Start the timer .

Function Call `spelling_suggestions` to find matches.

Print the input word and matching suggestions.

Step to Stop the timer.

Print the execution time.

- `dict2.json` (dictionary)

Github 10000 word Repo - <https://github.com/first20hours/google-10000-english>

Creating using regex manipulation command

```
find : ^(.+)$
```

```
replace : "$1",
```

## Code Demo

### Output 1

```
Enter the string/sentence: I defintely think we should recieve the package tommorow.
Input word: i
Matching word(s): [('i', 0)]
Input word: defintely
Matching word(s): [('definitely', 1)]
Input word: think
Input word: we
Matching word(s): [('we', 0)]
Input word: should
Matching word(s): [('should', 0)]
Input word: recieve
Matching word(s): [('receive', 2), ('believe', 2), ('recipe', 2), ('retrieve', 2)]
Input word: the
Matching word(s): [('the', 0)]
Input word: package
Matching word(s): [('package', 0)]
Input word: tommorow.
Matching word(s): [('tomorrow', 3)]
Execution time: 2.259100 seconds
```

### Output 2

```
Enter the string/sentence: The recived invitaton to the ceremony was mispalced, causing unnecessary confsion and delaays
Input word: the
Matching word(s): [('the', 0)]
Input word: recived
Matching word(s): [('received', 1)]
Input word: invitaton
Matching word(s): [('invitation', 1)]
Input word: to
Matching word(s): [('to', 0)]
Input word: the
Matching word(s): [('the', 0)]
Input word: ceremony
Matching word(s): [('ceremony', 0)]
Input word: was
Matching word(s): [('was', 0)]
Input word: mispalced,
Matching word(s): [('dispatched', 4), ('myspace', 4)]
Input word: causing
Matching word(s): [('causing', 0)]
Input word: unnecessary
Matching word(s): [('unnecessary', 1)]
Input word: confsion
Matching word(s): [('confusion', 1)]
Input word: and
Matching word(s): [('and', 0)]
Input word: delaays
Matching word(s): [('delays', 1)]
Execution time: 3.725358 seconds
```

### Word count:

1. "I defintely think we should recieve the package tommorow." — 9 words ( MS Copilot, 2024 )

Execution time: 2.175047 seconds

2. "The recived invitaton to the ceremony was mispalced, causing unnecesary confsion and delaays." — 13 words ( MS Copilot, 2024 )

Execution time: 3.725358 seconds

## Learning Experience

This assignment involved the working of dynamic programming for scoring spelling error where the initial program with Levenshtein Algorithm is implement in Python (version 3.8.2). Python library used for time calculation, using JSON dictionary and accessing Levenshtein Algorithm.

Through visualization and performing string operation through programming has enhanced the undestanding for Levenshtein Distance Equation.

### Levenshtein Distance Equation

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

(Tejaswi, 2024)

The application program for spelling suggestion provides is a small implementation for the this project. This program provides solution according to the edit distance and the corresponding data (dictionary). Meanwhile, lacks in providing 1 to 1 accurate answer due to its lack of literature grammar, data size and type error recognition. Developing on type error recognition by phonological errors - which happens when trying to spell, orthographic errors – when spelling contain incorrect pattern to original word (MW LITERACY, 2022) and typo errors – mistakes that are done by typing(Prasain, 2024).

Using a proper data base with English Dictionary words could enhance the word precision.

### References

- MW LITERACY. (2022, April 19). *The Three Types of Spelling Errors*. MW LITERACY®. <https://www.mw-literacy.com/blog/2022/4/17/the-three-types-of-spelling-errors>
- Prasain, B. (2024, March 2). *Types of spelling Errors - Bhawana Prasain - Medium*. Medium. <https://medium.com/@bhawana.prs/types-of-spelling-errors-de156517f226>
- Tejaswi, Y. (2024, April 20). *The Levenshtein Distance Algorithm: A string metric for measuring the difference between two sequences*. Medium. <https://medium.com/@tejaswiyadav221/the-levenshtein-distance-algorithm-a-string-metric-for-measuring-the-difference-between-two-269afbddd34>

MS COPILOT - make more spelling error for this sentence

Generated - The recived invitaton to the ceremony was mispalced, causing unnecessary confsion and delaays.

You

count words in "I defintely think we should recieve the package tommorow. "  
and "The recived invitaton to the ceremony was mispalced, causing unnecessary  
confsion and delaays."

Generated - The word count for each sentence is as follows:

"I defintely think we should recieve the package tommorow." — 9 words

"The recived invitaton to the ceremony was mispalced, causing unnecessary confsion  
and delaays." — 13 words