



COURSE NAME: Algorithm Analysis

STUDENT: Ertuğrul ŞENTÜRK

HOMEWORK SUBJECT: Design an algorithm for correcting misspelled words.

Algorithm:

1. Two hash table generated for dictionary and correction values. In correction table stored both original and corrected word. In dictionary table only the words in the smallDictionary.txt file added into the hash table.
2. An input was received from the user and the received input was divided word by word and saved in a array then each word searched in dictionary table. If it is in dictionary table word is unchanged.
3. If word is not in dictionary table then we searched the correction table. If word is in the correction table then it replaced with corrected form and moved into the next word.
4. If the word is not in both tables then it compared with all the words in dictionary table and calculated Levenshtein Edit Distance.
5. We only considered distance 1 and 2 words and stored them into the separate strings.
6. We printed the user if any of the word has distance 1 otherwise we printed with distance 2 words.
7. We get another input for corrected word from the user and stored it into the correction table and we also replaced it from the input text array.
8. Process iterated for each word.
9. The corrected text printed to user.

Levenshtein edit distance calculation algorithm up to 2 steps.

1. An empty matrix created.
2. Iterator moved into diagonally while both words have same characters.
3. If this process reaches the last cell, then returned 0.
4. Otherwise, we added 1 into the reached cell.
5. In that point we can move through to 3 different way. Right down or diagonally so we moved for all three options.
6. If the algorithm reaches the last cell, then we returned 1.
7. Otherwise, for each point we can move into 3 direction, so we moved into all 9 directions. If any of the calculation reaches the last cell, then we returned 2 otherwise we returned -1.
8. With that method we did not filled all cells

Screenshots for filling matrix:

	_	H	Y	U	N	D	A	I
_	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0
O	0	0	1	2	3	0	0	0
N	0	0	2	2	2	0	0	0
D	0	0	3	3	3	2	0	0
A	0	0	0	0	0	0	2	0

	_	H	Y	U	N	D	A	I
_	0	1	2	3	4	5	6	7
H	1	0	1	2	3	4	5	6
O	2	1	1	2	3	4	5	6
N	3	2	2	2	2	3	4	5
D	4	3	3	3	3	2	3	4
A	5	4	4	4	4	3	2	3

	_	S	T	R	E	A	M
_	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0
E	0	0	0	1	1	0	0
A	0	0	0	2	2	1	0
M	0	0	0	3	3	3	1

	_	S	T	R	E	A	M
_	0	1	2	3	4	5	6
S	1	0	1	2	3	4	5
T	2	1	0	1	2	3	4
E	3	2	1	1	1	2	3
A	4	3	2	2	2	1	2
M	5	4	3	3	3	2	1

Screenshots for application:

```
Welcome to the word deduction application
Please enter some text: I love icecream a lot
In word love did you mean?
- above or low or load or long or note or save or more or look or mode or like or code or line or have or one or lower
: like
In word lot did you mean?
- low : bit
I like icecream a bit

Press q to exit press enter to continue...
Please enter some text: I love icecream a lot
I like icecream a bit

Press q to exit press enter to continue...
```

```
Welcome to the word deduction application
Please enter some text: look at me please
In word me did you mean?
- e or be or we or my : my
look at my please

Press q to exit press enter to continue...
Please enter some text: LOOK AT ME PLEASE
LOOK AT my PLEASE

Press q to exit press enter to continue...
```

```
Please enter some text: here is too cold
In word here did you mean?
- there : there
In word too did you mean?
- to : to
In word cold did you mean?
- hold : held
there is to held

Press q to exit press enter to continue...
Please enter some text: here is too cold my love
In word love did you mean?
- above or low or load or long or note or save or more or
look or mode or like or code or line or have or one or lo
wer : code
there is to held my code

Press q to exit press enter to continue...
```