PLAGIARISM DETECTION BETWEEN TWO FILES

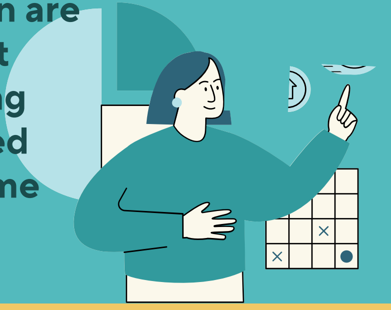
Longest Common Subsequence

PLAGIARISM VS LCS

The problem of academic plagiarism is addressed through the use of the *Longest Common Subsequence* (LCS) algorithm. With the aim of preserving academic integrity, addressing ethical concerns, and promoting academic excellence.

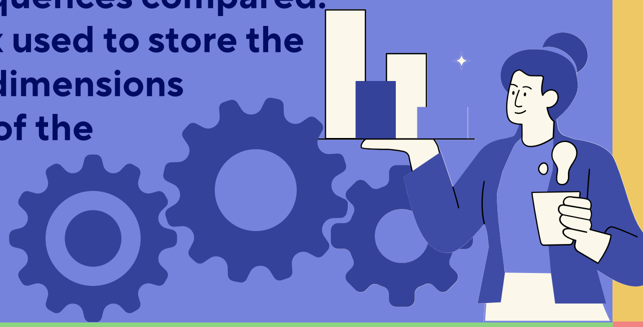
HOW LCS WORKS?

The LCS algorithm uses a matrix to compare characters from two sequences. It starts by creating an m*n matrix. where m and n are the lengths of the sequences. Then, it iterates through the matrix, comparing characters and updating lengths based on whether the characters are the same or different.

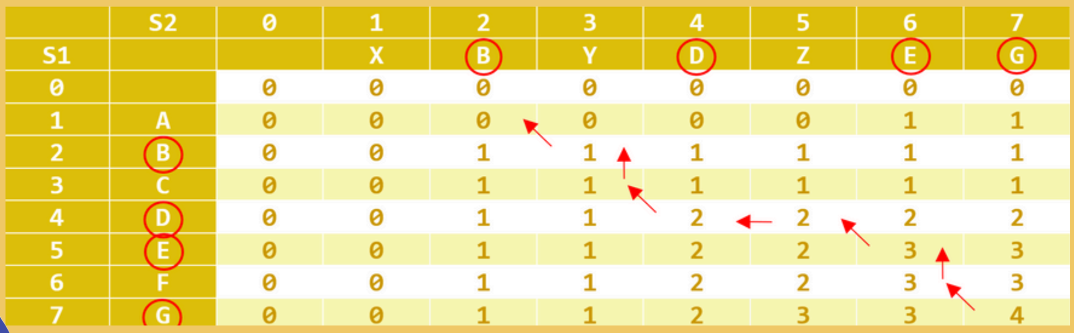


TIME COMPLEXITY

The complexity is O(m*n), where m and n are the lengths of the two sequences compared. This is because the matrix used to store the intermediate results has dimensions m+1 by n+1, and each cell of the matrix requires constant time to be calculated.



VISUALIZATION



	S2	0	1	2	3	4	5	6	7
S1			X	B	Y	D	Z	E	G
0		0	0	0	0	0	0	0	0
1	A	0	0	0	0	0	0	1	1
2	B	0	0	1	1	1	1	1	1
3	C	0	0	1	1	1	1	1	1
4	D	0	0	1	1	2	2	2	2
5	E	0	0	1	1	2	2	3	3
6	F	0	0	1	1	2	2	3	3
7	G	0	0	1	1	2	3	3	4

The LCS considered is BDEG, to our application it is EG.

LOOP INVARIANT

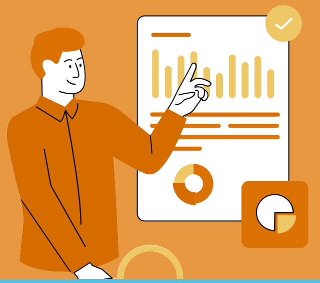
In each iteration of the main loop, the current submatrix contains the length of the longest common subsequence between the corresponding subsequences of the original sequences up to the current index.

APPLICATION

Only consecutive characters with a minimum length will be considered. The number of characters will be counted to get the percentage of plagiarism. Furthermore, different subsequences meeting these criteria will be stored, and their cumulative sum will provide the total plagiarism percentage, allowing users to judge if its plagiarism or not.

RECURRENCE RELATION

LCS[i][j] = 0, if [i] == 0 or [j] == 0
LCS[i][j] = LCS[i-1][j-1] + 1, if X[i] == Y[j]
LCS[i][j] = max(LCS[i-1][j], LCS[i][j-1]), if X[i] != Y[j]



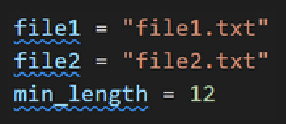
Where LCS[i][j] represents the length of the longest common subsequence between the first i elements of sequence X and the first j elements of sequence Y.

IN CONCLUSION...

The LCS algorithm offers an efficient method for determining the length of the longest common subsequence between two sequences. This algorithm is instrumental in various fields, including plagiarism detection, bioinformatics, and data compression, due to its ability to swiftly and accurately identify patterns of similarity.

LET'S CHECK IT OUT!

Based on the user's preferences, including the selection of parameters like the two files for comparison and a minimum number of characters for the longest common substring.



```
file1 = "file1.txt"
file2 = "file2.txt"
min_length = 12
```

Here is a preview of the two texts that are going to be compared:

```
text1 text1 text1 this is first plagiarism text1 text1 text1
text1 text1 text1 this is second plagiarism text1 text1 text1


text2 text2 text2 this is first plagiarism text2 text2 text2
text2 text2 text2 this is second plagiarism text2 text2 text2
```

Output results:

```
Identical text(s):
1.  this is second plagiarism text (24.8%)
2.  this is first plagiarism text (24.0%)
Percentage of identical text: 48.8%
```