String processing

1 - Typo suggestions

In a given search engine, an algorithm was developed to find words that are similar to the search query typed by the user. This allows the engine to suggest words that closely resemble accidental word mistypings.

The similarity between two words A and B is given by the edit/Levenshtein distance, which corresponds to the minimum amount of operations to convert A to B. The allowed operations are:

- Add a character to the word at any position.
- Remove a character from the word at any position.
- Swap any character at any position with any other character.

a) Indicate the recursive formula for the edit/Levenshtein distance.
b) Using a special case of the Needleman-Wunsch algorithm, compute the edit distance between “botao” and “abeto”.
c) Indicate a minimum ordered list of operations to convert “botao” to “abeto”.
d) When writing a search query, it is more common for a user to swap a character by mistake or forget to type a character, than to add a bogus character. Indicate the new recursive formula for the edit distance in the following scenarios:
    i) Adding a character has a weight of 2, while remove and swapping characters has a weight of 1.
    ii) Adding a character to the word is prohibited.

2 - Compacting messages

Different text compression strategies are being considered to send the following message through a pipeline: "pimpampumcadabolamataum":

a) Define a constant coding system for the text above. What is the minimum code size and the cost of encoding for the given text?
b) Determine the Huffman coding tree for this text, explaining in detail the whole process. What is the cost of coding in this case?
c) Using the Huffman tree calculated in the previous paragraph, present the codification of the phrase "pimpampum" and the encoding of each character.