DESIGN - Assignment 7 - The Great Firewall of Santa Cruz

The objective of this assignment is to provide a method of parsing through text and finding any words that overlap with a predetermined list of words. This is accomplished by using a Bloom filter, which is a bit vector that sets the bit through three different hash functions. If and when a word is searched in the Bloom filter, there's a strong possibility that the word is in the set of words listed. To make certain, you then have to check the hash table, and if the node is not in the table, then a false positive was listed.

Each node contains a string representing the old word, the new word, and its left and right neighbors. If there is no new word available, then the word is labeled as "badspeak" and will be noted as such.

The first step is making sure of your hash function:
Take in a number
Add to the number
Rotate the number, right shift circular
XOR the number
Return it as an index to an array of node pointers
If there's already a node at that index, add it as a BST

The hash table contains an array of varying length, and each index is filled with either a null pointer, or a pointer to a node that contains the old and new word. In the event of a hash collision, then a binary search tree will be created at that address, and that tree can be traversed in the event that the root node is ever looked up.

The second step is to fill the Bloom filter with the appropriate words:
Take in a word, and hash it three different times
Assign the bit vector with those values
Overlapping bits may occur

The distinction between the hash table and the Bloom filter is that the Bloom filter contains the possibility of whether or not the hash table has the word that was listed, and the hash table is more of an assurance. In theory, the Bloom filter would be unnecessary, as any hash collisions are resolved with the binary search tree, but that's besides the point.

You must then parse through the text:

Read in the text from standard input
Use regex to keep track of all proper characters, and chunk them
If the current word is in the Bloom filter,
    If the current word is in the hash table,
        Save the node for further use
    Else
        False positive

From this point, there are three separate actions that the program can take: it can either return a negative message, in which the person has used "badspeak", for which there is no translation, it can return a positive message, in which the person has used "oldspeak", for which there is a translation, or it can return a mixed message, which is a mixture of both. In addition to printing this message, the words that were flagged must be listed, in alphabetical order. To make this work, two separate binary search trees will be kept, that hold either all of the badspeak words or all of the oldspeak words, with their respective translations.

Check if the oldspeak tree is empty
Check if the badspeak tree is empty
Determine which message must be sent
Perform an inorder traversal of one or both trees
Print in alphabetical order, with the translation

Credit for the code provided and other helpful information goes to Professor Long and the staff of CSE-13S in the Fall 2021 quarter.