Solution Review: Total Number of Words in a Trie

This review provides a detailed analysis of the solution to the Total Number of Words in a Trie Challenge.

We'll cover the following

Solution: Increment Recursively

main.py

Trie.py

TrieNode.py

```python
from Trie import Trie
from TrieNode import TrieNode


# TrieNode => {children, is_end_word, char,
# mark_as_leaf(), unmark_as_leaf()}
def total_words(root):
    result = 0

    # Leaf denotes end of a word
    if root.is_end_word:
        result += 1

    for i in range(26):
        # Check if the node has children
        if root.children[i] is not None:
            # Recursively return the word count
            result += total_words(root.children[i])
    return result


keys = ["the", "a", "there", "answer", "any", "by", "bye", "th

trie = Trie()

for key in keys:
    trie.insert(key)

print(total_words(trie.root))

```

It's a pretty straightforward algorithm. Starting from the `root`, we visit each branch recursively. Whenever a node is found with its `isEndWord` set to `True`, the `result` variable is incremented by 1.

Time Complexity

For a trie with **n** number of nodes, the algorithm runs in *O(n)* because each node has to be traversed

Back

Challenge 1: Total Number of Words i...

Next

Challenge 2: Find All Words Stored in ...

✓ Mark as Completed