



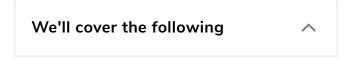






Solution Review: Total Number of Words in a Trie

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



- Solution: Increment Recursively
 - Time Complexity

Solution: Increment Recursively

```
main.py
Trie.py
TrieNode.py
    from Trie import Trie
    from TrieNode import TrieNode
  3
    # TrieNode => {children, is_end_word, char}
     def total_words(root):
  7
         result = 0
  8
  9
         # Leaf denotes end of a word
         if root.is_end_word:
 10
 11
              result += 1
 12
 13
         for letter in root.children:
```

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

Time Complexity

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

Interviewing soon? We've partnered with Hired so that companies apply to you instead of you applying to them. See how ①



Challenge 1: Total Number of Words i...

Challenge 2: Find All Words Stored in ...

