Solution Review: List Sort Using Trie

This review provides a detailed analysis of the solution to the Array Sort Using Tries Challenge.

We'll cover the following

Solution: Pre-Order Traversal

main.py

Trie.py

TrieNode.py

```
from Trie import Trie
from TrieNode import TrieNode
# Recursive Function to generate all words in alphabetic order


def get_words(root, result, level, word):
    # Leaf denotes end of a word
    if (root.is_end_word):
        # current word is stored till the 'level' in the chara
        temp = ""
        for x in range(level):
            temp += word[x]
        result.append(temp)

    for i in range(26):
        if (root.children[i] is not None):
            # Non-null child, so add that index to the charact
            word[level] = chr(i + ord('a'))
            get_words(root.children[i], result, level + 1, wor


def sort_list(arr):
    result = []

    # Creating Trie and Inserting words from array
    trie = Trie()
    for x in range(len(arr)):
        trie.insert(arr[x])

    word = [''] * 20
    get_words(trie.get_root(), result, 0, word)
    return result


keys = ["the", "a", "there", "answer", "any", "by", "bye", "th
print(sort_list(keys))

```

This exercise is very similar to **Challenge 2**, except the fact that you have to create the trie yourself.

Since the `children` list for each node stores letters in alphabetical order, the tree itself is ordered from top to bottom. All we need to do is make a pre-order traversal (think of `a` as the left most child and `z` as the right most child) and store the words in a list just like we did in the previous challenge.

Time Complexity

We first insert the nodes into the graph and then traverse all the existing nodes. Hence, the bottleneck worst case time complexity is *O(n)*.

Back

Challenge 3: List Sort Using Trie

Next

Challenge 4: Word Formation From a ...

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/solution-review-list-sort-using-trie__trie__data-structures-for-coding-interviews-in-python)