Challenge 3: List Sort Using Trie

This lesson is about sorting string lists using tries.

We'll cover the following

Problem Statement

In this problem, you have to implement the `sort_list()` function which will sort the elements of a list of strings.

Input

A list of strings.

Output

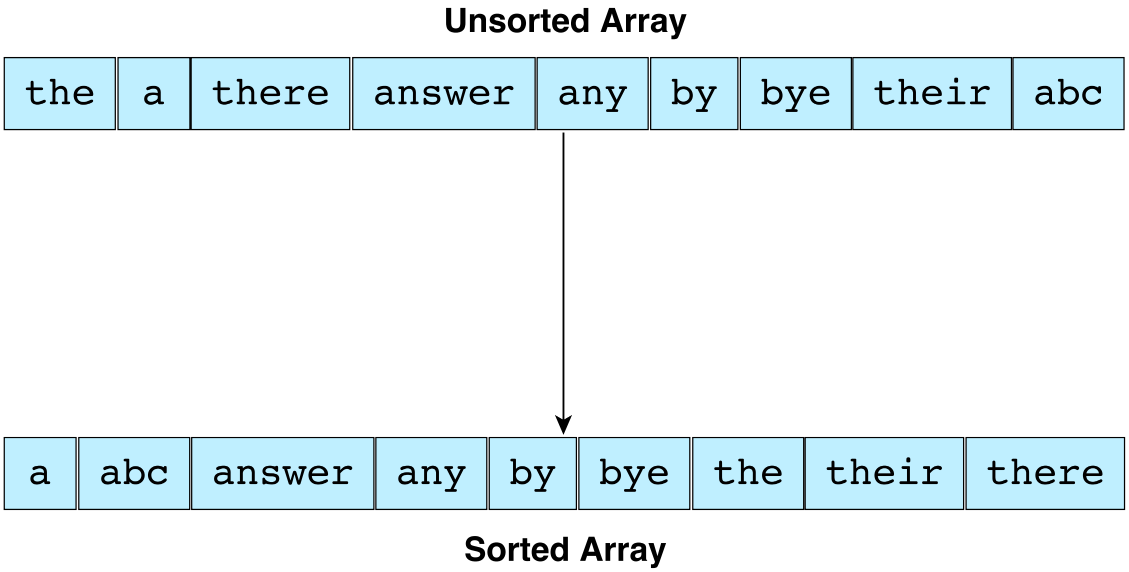
Returns the input list in a sorted state.

Sample Input

```
keys = ["the", "a", "there", "answer", "any",
                    "by", "bye", "their","abc"]
```

Sample Output

```
['a', 'abc','answer','any','by','bye','the','their','there']
```



Unsorted Array

the | a | there | answer | any | by | bye | their | abc

a | abc | answer | any | by | bye | the | their | there

Sorted Array

Coding Exercise

Carefully examine the information above and design a step-by-step algorithm first before jumping on to the implementation. This problem is designed for your practice, so try to solve it on your own first.

If you get stuck, you can always refer to the solution.

Good luck!

main.py

Trie.py

TrieNode.py

```
from Trie import Trie
from TrieNode import TrieNode


# Create Trie => trie = Trie()
# TrieNode => {children, is_end_word, char}
# Insert a Word => trie.insert(key)
# Search a Word => trie.search(key) return true or false
# Delete a Word => trie.delete(key)

# Recursive Function to generate all words in alphabetic order
def traverse(root, res, current):
    if root is None:
        return None
    current += root.char
    if root.is_end_word:
        res.append(current)
    for child in root.children:
        if child:
            traverse(child, res, current)
    return None
def sort_list(arr):
    trie = Trie()
    for word in arr:
        trie.insert(word)
    res = []
    _ = traverse(trie.root, res, "")
    return res
```

← Back

Solution Review: Find All Words Store...

Solution Review: List Sort Using Trie

☑ Completed

Report an Issue