Challenge 2: Find All Words Stored in Trie

If you are given a trie, can you return every word it contains?

We'll cover the following   ︿

- Problem Statement
  - Input
  - Output
  - Sample Input
  - Sample Output
- Coding Exercise

Problem Statement

You have to implement the `find_words()` function which will return a sorted list of all the words stored in a trie.

Input

The root node of a trie.

Output

A sorted list of all the words stored in a trie.   ☾
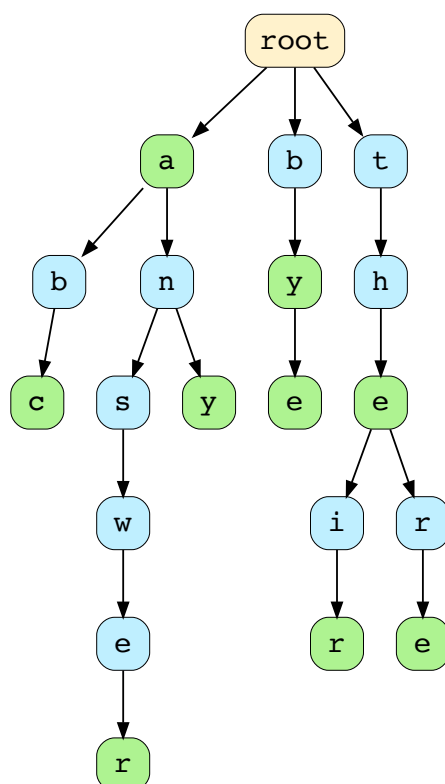
Sample Input

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

Sample Output

```
["a", "abc", "answer", "any", "by", "bye", "the", "their", "there"
] // Sorted alphabetically
```


Coding Exercise

Take a close look at 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 provided. We will discuss the solution in the Solution Review.

Good luck!

main.py

Trie.py

TrieNode.py

```python
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)
def _find_words(root, res, curr_word):
    if root is None:
        return res
    curr_word += root.char
    if root.is_end_word:
        res.append(curr_word)

    for child in root.children:
        if child:
            _find_words(child, res, curr_word)
    return None
def find_words(root):
    res = []
    _find_words(root, res, "")
    return res

    # Write your code here
    pass  # Return a list of strings
```

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

← **Back**

Solution Review: Total Number of Wor...

Next →

Solution Review: Find All Words Store...

✔ Completed

⚠ Report an Issue

🌙