

Data Structures

Trie Homework 2

Mostafa S. Ibrahim

Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada

Bachelor / Msc from Cairo University - Egypt

Ex-(Software Engineer / ICPC World Finalist)



Problem #1: Listing tree content

- `def` `get_all_strings(self)`
- Write a function that traverse a trie and **returns a list** of all the **complete** words in the trie
- The returned list should be **sorted** in alphabetical order

```
root = Trie()

root.insert("xyz")
root.insert("abcd")
root.insert("abf")
root.insert("ab")
root.insert("xyzwfe")

print(root.get_all_strings())
# ['ab', 'abcd', 'abf', 'xyz', 'xyzwfe']
```

Problem #2: Autocomplete

- `def autocomplete(self, str)`
- Given a string, the function will make an auto-complete for it, like google search
- In other words, it returns a list of all **full** words in the trie that has this string as prefix

```
root = Trie()
```

```
root.insert("xyz")  
root.insert("abcd")  
root.insert("abf")  
root.insert("ab")  
root.insert("xyzwfe")
```

```
print(root.autocomplete('a'))  
# ['ab', 'abcd', 'abf']
```

```
print(root.autocomplete('xy'))  
# ['xyz', 'xyzwfe']
```

Problem #3: [LeetCode 676](#) Implement Magic Dictionary

- `def word_exist_changes(self, str)`
- Returns true if, upon changing **exactly one** character in your input str, the resulting changed str matches **any full word** already in the trie. Otherwise, it returns false
- Assume the trie is: "hello", "leetcode"
- Input \Rightarrow output
 - hello \Rightarrow False [0 changes]
 - xello \Rightarrow True [1 change x to h]
 - xyllo \Rightarrow False [2 change to match the trie word]
- Assume the trie is: "hello", "leetcode", "hallo"
 - hello \Rightarrow True

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”