

Description

Solution

Discuss (999+)

Submissions

i

Java

Autocomplete

i

{ }

↺

↻

⌕

208. Implement Trie (Prefix Tree)

Medium 5209 77 Add to List Share

A **trie** (pronounced as "try") or **prefix tree** is a tree data structure used to efficiently store and retrieve keys in a dataset of strings. There are various applications of this data structure, such as autocomplete and spellchecker.

Implement the Trie class:

- Trie() Initializes the trie object.
- void insert(String word) Inserts the string word into the trie.
- boolean search(String word) Returns true if the string word is in the trie (i.e., was inserted before), and false otherwise.
- boolean startsWith(String prefix) Returns true if there is a previously inserted string word that has the prefix prefix, and false otherwise.

Example 1:

Input

["Trie", "insert", "search", "search", "startsWith", "insert", "search"]

[[], ["apple"], ["apple"], ["app"], ["app"], ["app"], ["app"]]

Output

[null, null, true, false, true, null, true]

Explanation

Trie trie = new Trie();
trie.insert("apple");
trie.search("apple"); // return True
trie.search("app"); // return False
trie.startsWith("app"); // return True
trie.insert("app");
trie.search("app"); // return True

Constraints:

- 1 <= word.length, prefix.length <= 2000
- word and prefix consist only of lowercase English letters.
- At most 3 * 10⁴ calls in total will be made to insert, search, and startsWith.

Accepted 457,633 Submissions 839,072

Seen this question in a real interview before?

Yes

No

Companies

Related Topics

Similar Questions

```
1 class Trie {
2
3     /** Initialize your data structure here. */
4     public Trie() {
5
6     }
7
8     /** Inserts a word into the trie. */
9     public void insert(String word) {
10
11     }
12
13     /** Returns if the word is in the trie. */
14     public boolean search(String word) {
15
16     }
17
18     /** Returns if there is any word in the trie that starts with the given prefix. */
19     public boolean startsWith(String prefix) {
20
21     }
22 }
23
24 /**
25  * Your Trie object will be instantiated and called as such:
26  * Trie obj = new Trie();
27  * obj.insert(word);
28  * boolean param_2 = obj.search(word);
29  * boolean param_3 = obj.startsWith(prefix);
30  */
```