Description Solution Discuss (106) Submissions

1804. Implement Trie II (Prefix Tree)

Medium 107 7 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.
- `int countWordsEqualTo(String word)` Returns the number of instances of the string `word` in the trie.
- `int countWordsStartingWith(String prefix)` Returns the number of strings in the trie that have the string `prefix` as a prefix.
- `void erase(String word)` Erases the string `word` from the trie.

Example 1:

```
Input
["Trie", "insert", "insert", "countWordsEqualTo",
"countWordsStartingWith", "erase", "countWordsEqualTo",
"countWordsStartingWith", "erase", "countWordsStartingWith"]
[[], ["apple"], ["apple"], ["apple"], ["app"], ["apple"],
["apple"], ["app"], ["apple"], ["app"]]
Output
[null, null, null, 2, 2, null, 1, 1, null, 0]

Explanation
Trie trie = new Trie();
trie.insert("apple");               // Inserts "apple".
trie.insert("apple");               // Inserts another
"apple".
trie.countWordsEqualTo("apple");    // There are two instances
of "apple" so return 2.
trie.countWordsStartingWith("app"); // "app" is a prefix of
"apple" so return 2.
trie.erase("apple");                // Erases one "apple".
trie.countWordsEqualTo("apple");    // Now there is only one
instance of "apple" so return 1.
trie.countWordsStartingWith("app"); // return 1
trie.erase("apple");                // Erases "apple". Now the
trie is empty.
trie.countWordsStartingWith("app"); // return 0
```

Constraints:

- `1 <= word.length, prefix.length <= 2000`
- `word` and `prefix` consist only of lowercase English letters.
- At most $3 * 10^4$ calls **in total** will be made to `insert`, `countWordsEqualTo`, `countWordsStartingWith`, and `erase`.
- It is guaranteed that for any function call to `erase`, the string `word` will exist in the trie.

Accepted 4,863 Submissions 8,385

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

Show Hint 1

Show Hint 2

Show Hint 3

Java Autocomplete

```java
class Trie {

    public Trie() {

    }

    public void insert(String word) {

    }

    public int countWordsEqualTo(String word) {

    }

    public int countWordsStartingWith(String prefix) {

    }

    public void erase(String word) {

    }
}

/**
 * Your Trie object will be instantiated and called as such:
 * Trie obj = new Trie();
 * obj.insert(word);
 * int param_2 = obj.countWordsEqualTo(word);
 * int param_3 = obj.countWordsStartingWith(prefix);
 * obj.erase(word);
 */
```