

211. Design Add and Search Words Data Structure

Medium 3419 143 Add to List Share

Design a data structure that supports adding new words and finding if a string matches any previously added string.

Implement the `WordDictionary` class:

- `WordDictionary()` Initializes the object.
- `void addWord(word)` Adds `word` to the data structure, it can be matched later.
- `bool search(word)` Returns `true` if there is any string in the data structure that matches `word` or `false` otherwise. `word` may contain dots `'.'` where dots can be matched with any letter.

Example:

Input
["WordDictionary", "addWord", "addWord", "addWord", "search", "search",

[[], ["bad"], ["dad"], ["mad"], ["pad"], ["bad"], [".ad"], ["b.."]]
Output
[null, null, null, null, false, true, true, true]





Explanation
WordDictionary wordDictionary = new WordDictionary();
wordDictionary.addWord("bad");
wordDictionary.addWord("dad");
wordDictionary.addWord("mad");
wordDictionary.search("pad"); // return False
wordDictionary.search("bad"); // return True
wordDictionary.search(".ad"); // return True
wordDictionary.search("b.."); // return True

Constraints:

- `1 <= word.length <= 500`
- `word` in `addWord` consists lower-case English letters.
- `word` in `search` consist of `'.'` or lower-case English letters.
- At most `50000` calls will be made to `addWord` and `search`.

Accepted 314,991 Submissions 749,732

Seen this question in a real interview before?

- Companies  
- Related Topics 
- Similar Questions 
- Show Hint 1 

```
1 class WordDictionary {
2
3     /** Initialize your data structure here. */
4     public WordDictionary() {
5
6     }
7
8     public void addWord(String word) {
9
10    }
11
12    public boolean search(String word) {
13
14    }
15 }
16
17 /**
18  * Your WordDictionary object will be instantiated and called as such:
19  * WordDictionary obj = new WordDictionary();
20  * obj.addWord(word);
21  * boolean param_2 = obj.search(word);
22  */
```