Problem 244: Shortest Word Distance II

Problem Information

Difficulty: Medium
Acceptance Rate: 62.50%
Paid Only: Yes
Tags: Array, Hash Table, Two Pointers, String, Design

Problem Description

Design a data structure that will be initialized with a string array, and then it should answer queries of the shortest distance between two different strings from the array.

Implement the `WordDistance` class:

* `WordDistance(String[] wordsDict)` initializes the object with the strings array `wordsDict`. * `int shortest(String word1, String word2)` returns the shortest distance between `word1` and `word2` in the array `wordsDict`.

Example 1:

Input ["WordDistance", "shortest", "shortest"] [[["practice", "makes", "perfect", "coding", "makes"]], ["coding", "practice"], ["makes", "coding"]] **Output** [null, 3, 1] **Explanation** WordDistance wordDistance = new WordDistance(["practice", "makes", "perfect", "coding", "makes"]); wordDistance.shortest("coding", "practice"); // return 3 wordDistance.shortest("makes", "coding"); // return 1

Constraints:

* `1 <= wordsDict.length <= 3 * 104` * `1 <= wordsDict[i].length <= 10` * `wordsDict[i]` consists of lowercase English letters. * `word1` and `word2` are in `wordsDict`. * `word1 != word2` * At most `5000` calls will be made to `shortest`.

Code Snippets

C++:

```
class WordDistance {
public:
WordDistance(vector<string>& wordsDict) {

}

int shortest(string word1, string word2) {

}
};

/**
* Your WordDistance object will be instantiated and called as such:
* WordDistance* obj = new WordDistance(wordsDict);
* int param_1 = obj->shortest(word1,word2);
*/
```

Java:

```
class WordDistance {

public WordDistance(String[] wordsDict) {

}

public int shortest(String word1, String word2) {

}
}

/**
* Your WordDistance object will be instantiated and called as such:
* WordDistance obj = new WordDistance(wordsDict);
* int param_1 = obj.shortest(word1,word2);
*/
```

Python3:

```
class WordDistance:

def __init__(self, wordsDict: List[str]):
```

```
def shortest(self, word1: str, word2: str) -> int:



# Your WordDistance object will be instantiated and called as such:
# obj = WordDistance(wordsDict)
# param_1 = obj.shortest(word1,word2)
```