

Problem 804: Unique Morse Code Words

Problem Information

Difficulty: Easy

Acceptance Rate: 83.45%

Paid Only: No

Tags: Array, Hash Table, String

Problem Description

International Morse Code defines a standard encoding where each letter is mapped to a series of dots and dashes, as follows:

* 'a' maps to "-.", * 'b' maps to "-...", * 'c' maps to "-..-", and so on.

For convenience, the full table for the '26' letters of the English alphabet is given below:

Given an array of strings `words` where each word can be written as a concatenation of the Morse code of each letter.

* For example, ``cab'' can be written as ``-.-.-...'', which is the concatenation of ``-.-'', ``.-'', and ``-...''. We will call such a concatenation the ****transformation**** of a word.

Return the number of different**transformations** among all words we have ..

Example 1:

****Input:**** words = ["gin", "zen", "gig", "msg"] ****Output:**** 2 ****Explanation:**** The transformation of each word is: "gin" -> "--...-." "zen" -> "--...-." "gig" -> "--...--." "msg" -> "--...--.". There are 2 different transformations: "--...-." and "--...--.".

****Example 2:****

****Input:**** words = ["a"] ****Output:**** 1

****Constraints:****

* `1 <= words.length <= 100` * `1 <= words[i].length <= 12` * `words[i]` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {  
public:  
    int uniqueMorseRepresentations(vector<string>& words) {  
  
    }  
};
```

Java:

```
class Solution {  
public int uniqueMorseRepresentations(String[] words) {  
  
}  
}
```

Python3:

```
class Solution:  
    def uniqueMorseRepresentations(self, words: List[str]) -> int:
```