804 Unique Morse Code Words

2018年5月5日 17:12

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:
[".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."]
Now, given a list of words, 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 "-.-." + "-..." + ".-"). We'll call such a concatenation, the transformation of a word.
Return the number of different transformations among all words we have.
Example:
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 "--...--.".

Note:

- The length of words will be at most 100.
- Each words[i] will have length in range [1, 12].
- words[i] will only consist of lowercase letters.

来自 <https://leetcode.com/problems/unique-morse-code-words/description/>
国际摩尔斯密码定义一种标准编码方式，将每个字母对应于一个由一系列点和短线组成的字符串，比如: "a" 对应 ".-", "b" 对应 "-...", "c" 对应 "-.-.", 等等。
为了方便，所有26个英文字母对应摩尔斯密码表如下:
[".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."]
给定一个单词列表，每个单词可以写成每个字母对应摩尔斯密码的组合。例如，"cab" 可以写成 "-.-.-....-"，(即 "-.-." + "-..." + ".-"字符串的结合)。我们将这样一个连接过程称作单词翻译。
返回我们可以获得所有词不同单词翻译的数量。
例如:
输入: words = ["gin", "zen", "gig", "msg"]
输出: 2
解释:
各单词翻译如下:
"gin" -> "--...-."
"zen" -> "--...-."
"gig" -> "--...--."
"msg" -> "--...--."
共有 2 种不同翻译, "--...-." 和 "--...--.".
注意:

- 单词列表words 的长度不会超过 100。
- 每个单词 words[i]的长度范围为 [1, 12]。
- 每个单词 words[i]只包含小写字母。

Solution for Python3:

```python
class Solution:
    def uniqueMorseRepresentations(self, words):
        """
        :type words: List[str]
        :rtype: int
        """
        Morse = [".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."]
        seen = {"".join(Morse[ord(c) - ord('a')] for c in word) for word in words}
        return len(seen)
```

Solution for C++:

```cpp
class Solution {
public:
    int uniqueMorseRepresentations(vector<string>& words) {
        vector<string> Morse = {".-","-...","-.-.","-..",".","..-.","--.","....","..",".---","-.-",".-..","--","-.","---",".--.","--.-",".-.","...","-","..-","...-",".--","-..-","-.--","--.."};
        unordered_set<string> seen;
        for (string word : words) {
            string s = "";
            for (char c : word)
                s += Morse[c - 'a'];
            seen.insert(s);
        }
        return seen.size();
    }
};
```