**[Unique Morse Code Words](https://leetcode.com/problems/unique-morse-code-words/)**

**package** leetcodeEasy;

**import** java.util.HashSet;

**import** java.util.Set;

**public** **class** MorseCode {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String[] words = {"gin", "zen", "gig", "msg"};

System.***out***.println(*uniqueMorseRepresentations*(words));

}

**public** **static** **int** uniqueMorseRepresentations(String[] words) {

String[] morse = {".-","-...","-.-.","-..",".","..-.",

"--.","....","..",".---","-.-",".-..",

"--","-.","---",".--.","--.-",".-.","...",

"-","..-","...-",".--","-..-","-.--","--.."};

Set<String> set = **new** HashSet<>();

**for**(String word : words) {

String result = "";

**for**(**char** c : word.toCharArray()) {

result += morse[c - 'a'];

}

set.add(result);

}

**return** set.size();

}

}

Time Complexity : O(n\*l) , n is the number of elements in words array and l is average length of each word

Space Complexity : O(n), n is the number of elements in words array