



链滴

# Keyboard Row--leetcode

作者: [moloe](#)

原文链接: <https://ld246.com/article/1508842329418>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 题目

Given a List of words, return the words that can be typed using letters of **alphabet** on only one row's of American keyboard like the image below.

**Example:**

**Input:** ["Hello", "Alaska", "Dad", "Peace"]

**Output:** ["Alaska", "Dad"]

总的来说，这类问题，如果直接暴力解决，可能时间复杂度不太能满足。但是从题目来看，至少两轮历是需要的。整体上是要考虑减少一轮遍历。

考虑到可以通过每个单词的第一个字符来判断后面的字符由哪行来继续判断，这样就可以少遍历一。优化一下代码思路，`discuss` 中有个方案，就是用`map`存储每个原始字符的 row，后面取起来就会易一些。

```
class Solution {  
    public String[] findWords(String[] words) {  
        List<String> res = new ArrayList<>();  
  
        String[] lines = {"qwertyuiop", "asdfghjkl", "zxcvbnm"};  
  
        Map<Character, Integer> map = new HashMap<>();  
  
        // 把row index存下来  
        for(int i=0;i<lines.length;i++) {  
            for(int j=0;j<lines[i].toCharArray().length;j++) {  
                map.put(lines[i].charAt(j), i);  
            }  
        }  
  
        // 遍历每个word，根据首字符的row index来继续判断后面的字符  
        for(String word:words) {  
            int firstIndex = map.get(word.toLowerCase().charAt(0));  
            for (Character c:word.toLowerCase().toCharArray()) {  
                if(lines[firstIndex].indexOf(c) == -1) {  
                    firstIndex = -1;  
                    break;  
                }  
            }  
            if(firstIndex != -1) {  
                System.out.println(word);  
                res.add(word);  
            }  
        }  
        // list to string Array  
        return res.toArray(new String[0]);  
    }  
}
```