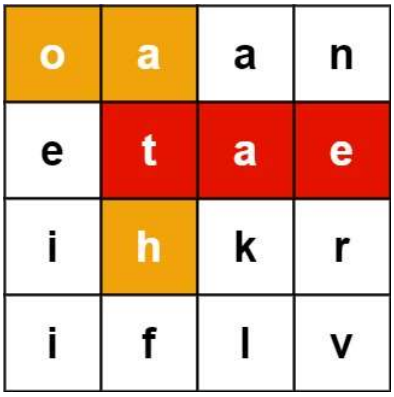
212. Word Search II

Hard 7893 366 Add to List Share

Given an `m x n board` of characters and a list of strings `words`, return *all words on the board*.

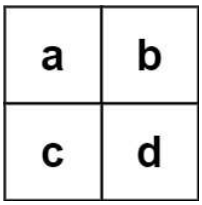
Each word must be constructed from letters of sequentially adjacent cells, where **adjacent cells** are horizontally or vertically neighboring. The same letter cell may not be used more than once in a word.

Example 1:




```
Input: board = [["o","a","a","n"],["e","t","a","e"],
["i","h","k","r"],["i","f","l","v"]], words =
["oath","pea","eat","rain"]
Output: ["eat","oath"]
```

Example 2:




```
Input: board = [["a","b"],["c","d"]], words = ["abcb"]
Output: []
```

Constraints:

- `m == board.length`
- `n == board[i].length`
- `1 <= m, n <= 12`
- `board[i][j]` is a lowercase English letter.
- `1 <= words.length <= 3 * 10^4`
- `1 <= words[i].length <= 10`
- `words[i]` consists of lowercase English letters.
- All the strings of `words` are unique.

Accepted 526,689 Submissions 1,427,278

Seen this question in a real interview before? Yes No

Companies

Related Topics

Similar Questions

Show Hint 1

Show Hint 2

Java Autocomplete

```
class Solution {
    public List<String> findWords(char[][] board, String[] words) {

    }
}
```