Word Search II

Try to solve the Word Search II problem.

We'll cover the following

Statement

You are given a list of strings that you need to find in a 2D grid of letters such that the string can be constructed from letters in sequentially adjacent cells. The cells are considered sequentially adjacent when they are neighbors to each other either horizontally or vertically. The solution should return a list containing the strings from the input list that were found in the grid.

Constraints:

- $1 \leq$ rows, columns ≤ 12
- $1 \leq$ `words.length` $\leq 3 \times 10^3$
- $1 \leq$ `words[i].length` ≤ 10
- All the strings are unique.

> **Note:** The order of the strings in the output does *not* matter.

Examples

Words to search: [REINDEER, IN, RAIN]

C	O	L	I	M
I	N	L	M	O
A	L	I	E	O
R	T	A	S	N
S	I	T	A	C

1 of 5

Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

Word Search II

1 From the given list of words, which words are present in the grid?

$$\begin{matrix} O & A & A & C \\ E & T & A & E \\ I & H & K & S \\ I & Z & L & D \end{matrix}$$

$$[OATH, PEA, EAT]$$

A) [EAT, OATH]

B) [EAT, PEA]

C) [OATH, PEA]

D) [PEA]

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

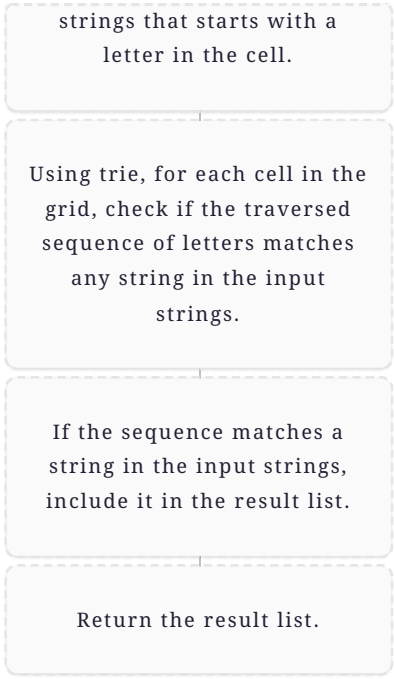
Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

Drag and drop the cards to rearrange them in the correct sequence.

Insert all the input strings in the trie.

Search through the grid cells, looking in all four possible adjacent directions to see if there is a string in the input



Reset | Show Solution | Submit

Try it yourself

Implement your solution in `WordSearch.java` in the following coding playground. We have provided useful code templates in the other files that you may build on to solve this problem.

WordSearch.java
Trie.java
TrieNode.java

```
1  import java.util.*;
2  class WordSearch {
3      public static List<String> findStrings(char[][] grid, String[] words){
4          // grid is a 2D array of characters
5          // strings is the list of strings that we are searching for
6          // write your code here
7          // your code will replace the return place holder
8
9          return new ArrayList<String>();
10     }
11 }
12
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

[["C","S","L","I","M"],["O","I","L","M","O"],["O","L","I","E","O"],["R","T","A","S","N"],["S","I","T","A","C"]]

Input #2

["SLIME","SAILOR","MATCH","COCOON"]

Back
Solution: Design Add ...

Next
Solution: Word Search...

Mark as Completed