Problem 425: Word Squares

Problem Information

Difficulty: Hard
Acceptance Rate: 54.53%
Paid Only: Yes
Tags: Array, String, Backtracking, Trie

Problem Description

Given an array of **unique** strings `words`, return _all the_**[word squares](https://en.wikipedia.org/wiki/Word_square)** _you can build from_`words`. The same word from `words` can be used **multiple times**. You can return the answer in **any order**.

A sequence of strings forms a valid **word square** if the `kth` row and column read the same string, where `0 <= k < max(numRows, numColumns)`.

* For example, the word sequence `["ball","area","lead","lady"]` forms a word square because each word reads the same both horizontally and vertically.

Example 1:

Input: words = ["area","lead","wall","lady","ball"] **Output:** [["ball","area","lead","lady"],["wall","area","lead","lady"]] **Explanation:** The output consists of two word squares. The order of output does not matter (just the order of words in each word square matters).

Example 2:

Input: words = ["abat","baba","atan","atal"] **Output:** [["baba","abat","baba","atal"],["baba","abat","baba","atan"]] **Explanation:** The output consists of two word squares. The order of output does not matter (just the order of words in each word square matters).

Constraints:

* `1 <= words.length <= 1000` * `1 <= words[i].length <= 4` * All `words[i]` have the same length. * `words[i]` consists of only lowercase English letters. * All `words[i]` are **unique**.

Code Snippets

C++:

```cpp
class Solution {
public:
vector<vector<string>> wordSquares(vector<string>& words) {

}
};
```

Java:

```java
class Solution {
public List<List<String>> wordSquares(String[] words) {

}
}
```

Python3:

```python
class Solution:
def wordSquares(self, words: List[str]) -> List[List[str]]:
```