[Click on the logo to find **Problem Statement**]

Intuition

The code aims to find the first palindrome in a given list of words. It iterates through each word in the list and checks if the word is equal to its reverse, indicating it is a palindrome. If a palindrome is found, it is returned immediately. If no palindrome is found after iterating through all words, an empty string is returned. This approach efficiently identifies the first palindrome without needing to iterate through the entire list in cases where a palindrome is found early.

Approach

1. Iterate through each word in the input list.
2. For each word, check if it is equal to its reverse. This can be done by comparing the word with its reversed version using slicing (`i[::-1]`).
3. If a word is found that is equal to its reverse, it means it's a palindrome. Return this palindrome immediately.
4. If no palindrome is found after iterating through all words, return an empty string indicating no palindrome exists in the list.

Complexity

- Time complexity:O(m*n)
- Space complexity: O(1)

Code

```python
class Solution:
    def firstPalindrome(self, words: List[str]) -> str:
        for i in words:
            if i == i[::-1]:
                return i
        return ""
```

If you want to see more solutions to coding problems, you can visit: