Palindrome Pair

Problem Level: Hard

Problem Description:

Given 'n' number of words, you need to find if there exist any two words which can be joined to make a palindrome or any word, which itself is a palindrome.
The function should return either true or false. You don't have to print anything.

Sample Input 1:

```
4
abc def ghi cba
```

Sample Output 1:

```
true
```

Explanation:

"abc" and "cba" forms a palindrome

Approach to be followed:

The approach we have taken to solve this problem is to store the reverse of each of the words in the Trie. We then iterate over the words and search whether the Trie contains the same word or not. It may happen that some parts of the word or a substring exist in the Trie. We check for the remaining part of the string to be a palindrome or not.

Vise versa of the above will also be true, that means, it may happen that the word in the Trie may extend further over different numbers of branches. Hence we check all the branches one by one to see if any of the branches make a palindrome.

Steps:

1. Using DFS, traverse the entire tree until the point left and right subtrees do not become null.
2. While backtracking
    a. Check if K is greater than 0, decrement it by 1.

    b. Else if K equals 0, ancestor is found in the current root's data.
3. If not found, ancestor becomes -1, representing it does not exist.
4. Base Case: if root equals null, return null and exit DFS.

Time Complexity: O(N * M), where **N** is the number of words in the Trie and **M** is the average word length.