Problem 1684: Count the Number of Consistent Strings

Problem Information

Difficulty: Easy
Acceptance Rate: 88.39%
Paid Only: No
Tags: Array, Hash Table, String, Bit Manipulation, Counting

Problem Description

You are given a string `allowed` consisting of **distinct** characters and an array of strings `words`. A string is **consistent** if all characters in the string appear in the string `allowed`.

Return _the number of**consistent** strings in the array _`words`.

Example 1:

Input: allowed = "ab", words = ["ad","bd","aaab","baa","badab"] **Output:** 2 **Explanation:** Strings "aaab" and "baa" are consistent since they only contain characters 'a' and 'b'.

Example 2:

Input: allowed = "abc", words = ["a","b","c","ab","ac","bc","abc"] **Output:** 7 **Explanation:** All strings are consistent.

Example 3:

Input: allowed = "cad", words = ["cc","acd","b","ba","bac","bad","ac","d"] **Output:** 4 **Explanation:** Strings "cc", "acd", "ac", and "d" are consistent.

Constraints:

* `1 <= words.length <= 104` * `1 <= allowed.length <= 26` * `1 <= words[i].length <= 10` * The characters in `allowed` are **distinct**. * `words[i]` and `allowed` contain only lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int countConsistentStrings(string allowed, vector<string>& words) {

}
};
```

Java:

```
class Solution {
public int countConsistentStrings(String allowed, String[] words) {

}
}
```

Python3:

```
class Solution:
def countConsistentStrings(self, allowed: str, words: List[str]) -> int:
```