Problem 1178: Number of Valid Words for Each Puzzle

Problem Information

Difficulty: Hard
Acceptance Rate: 47.55%
Paid Only: No
Tags: Array, Hash Table, String, Bit Manipulation, Trie

Problem Description

With respect to a given `puzzle` string, a `word` is _valid_ if both the following conditions are satisfied:

* `word` contains the first letter of `puzzle`. * For each letter in `word`, that letter is in `puzzle`. * For example, if the puzzle is `"abcdefg"`, then valid words are `"faced"`, `"cabbage"`, and `"baggage"`, while * invalid words are `"beefed"` (does not include `'a'`) and `"based"` (includes `'s'` which is not in the puzzle).

Return _an array_`answer` _, where_`answer[i]`_is the number of words in the given word list_`words` _that is valid with respect to the puzzle_`puzzles[i]`.

Example 1:

Input: words = ["aaaa","asas","able","ability","actt","actor","access"], puzzles = ["aboveyz","abrodyz","abslute","absoryz","actresz","gaswxyz"] **Output:** [1,1,3,2,4,0] **Explanation:** 1 valid word for "aboveyz" : "aaaa" 1 valid word for "abrodyz" : "aaaa" 3 valid words for "abslute" : "aaaa", "asas", "able" 2 valid words for "absoryz" : "aaaa", "asas" 4 valid words for "actresz" : "aaaa", "asas", "actt", "access" There are no valid words for "gaswxyz" cause none of the words in the list contains letter 'g'.

Example 2:

Input: words = ["apple","pleas","please"], puzzles = ["aelwxyz","aelpxyz","aelpsxy","saelpxy","xaelpsy"] **Output:** [0,1,3,2,0]

Constraints:

* `1 <= words.length <= 105` * `4 <= words[i].length <= 50` * `1 <= puzzles.length <= 104` * `puzzles[i].length == 7` * `words[i]` and `puzzles[i]` consist of lowercase English letters. * Each `puzzles[i] `does not contain repeated characters.

Code Snippets

C++:

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

}
};
```

Java:

```
class Solution {
public List<Integer> findNumOfValidWords(String[] words, String[] puzzles) {

}
}
```

Python3:

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