Problem 1170: Compare Strings by Frequency of the Smallest Character

Problem Information

Difficulty: Medium
Acceptance Rate: 63.05%
Paid Only: No
Tags: Array, Hash Table, String, Binary Search, Sorting

Problem Description

Let the function `f(s)` be the **frequency of the lexicographically smallest character** in a non-empty string `s`. For example, if `s = "dcce"` then `f(s) = 2` because the lexicographically smallest character is `'c'`, which has a frequency of 2.

You are given an array of strings `words` and another array of query strings `queries`. For each query `queries[i]`, count the **number of words** in `words` such that `f(queries[i])` < `f(W)` for each `W` in `words`.

Return _an integer array_`answer` _, where each_`answer[i]`_is the answer to the_`ith` _query_.

Example 1:

Input: queries = ["cbd"], words = ["zaaaz"] **Output:** [1] **Explanation:** On the first query we have f("cbd") = 1, f("zaaaz") = 3 so f("cbd") < f("zaaaz").

Example 2:

Input: queries = ["bbb","cc"], words = ["a","aa","aaa","aaaa"] **Output:** [1,2] **Explanation:** On the first query only f("bbb") < f("aaaa"). On the second query both f("aaa") and f("aaaa") are both > f("cc").

Constraints:

* `1 <= queries.length <= 2000` * `1 <= words.length <= 2000` * `1 <= queries[i].length, words[i].length <= 10` * `queries[i][j]`, `words[i][j]` consist of lowercase English letters.

Code Snippets

C++:

```cpp
class Solution {
public:
vector<int> numSmallerByFrequency(vector<string>& queries, vector<string>&
words) {

}
};
```

Java:

```java
class Solution {
public int[] numSmallerByFrequency(String[] queries, String[] words) {

}
}
```

Python3:

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