1178. Number of Valid Words for Each Puzzle

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 <= 10^5`
- `4 <= words[i].length <= 50`
- `1 <= puzzles.length <= 10^4`
- `puzzles[i].length == 7`
- `words[i]` and `puzzles[i]` consist of lowercase English letters.
- Each `puzzles[i]` does not contain repeated characters.