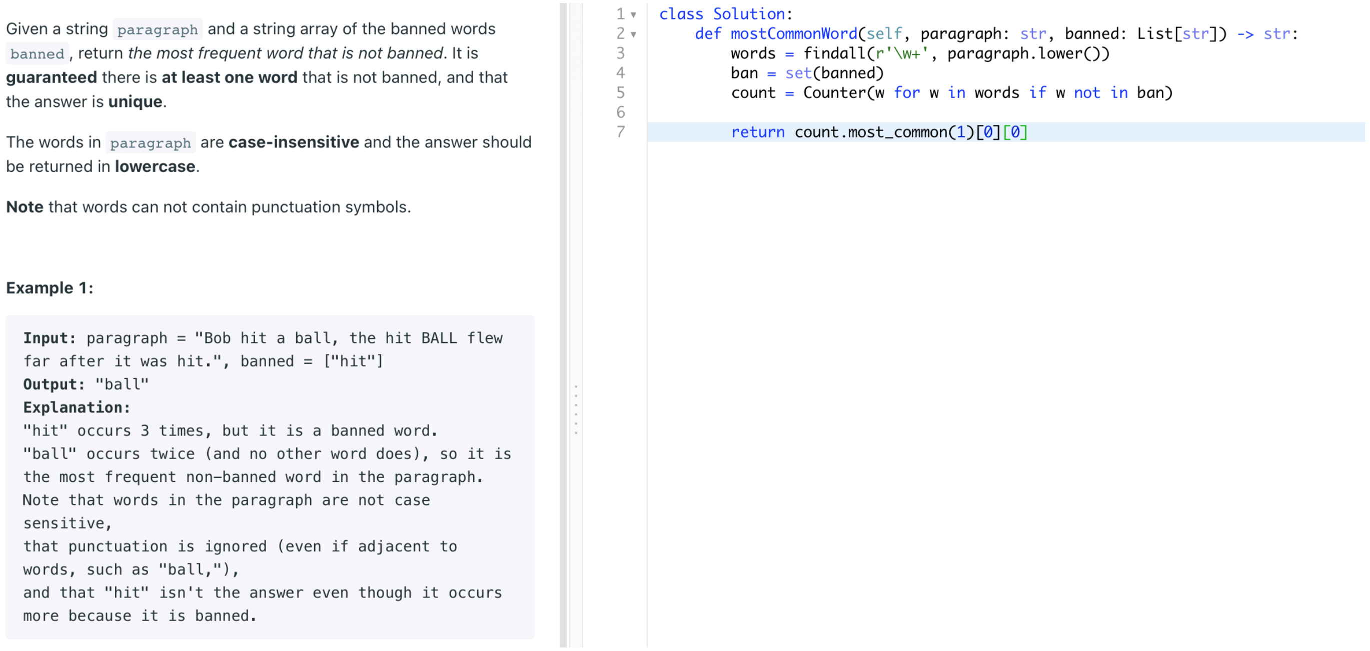

Given a string `paragraph` and a string array of the banned words `banned`, return *the most frequent word that is not banned*. It is **guaranteed** there is **at least one word** that is not banned, and that the answer is **unique**.

The words in `paragraph` are **case-insensitive** and the answer should be returned in **lowercase**.

**Note** that words can not contain punctuation symbols.

**Example 1:**

```
Input: paragraph = "Bob hit a ball, the hit BALL flew
far after it was hit.", banned = ["hit"]
Output: "ball"
Explanation:
"hit" occurs 3 times, but it is a banned word.
"ball" occurs twice (and no other word does), so it is
the most frequent non-banned word in the paragraph.
Note that words in the paragraph are not case
sensitive,
that punctuation is ignored (even if adjacent to
words, such as "ball,"),
and that "hit" isn't the answer even though it occurs
more because it is banned.
```

```python
class Solution:
    def mostCommonWord(self, paragraph: str, banned: List[str]) -> str:
        words = findall(r'\w+', paragraph.lower())
        ban = set(banned)
        count = Counter(w for w in words if w not in ban)

        return count.most_common(1)[0][0]
```

- \w+ captures alphanumeric words

- .most_common(1)[0][0]

  ↳ return list of top 1 frequent items

  Eg. c = { 'a': 1, 'b': 2, 'c': 3, 'd': 3 }

c.most_common(1) = [('c', 3), ('d', 3)]

1st [0]: grab 1st tuple in the list: ('c', 3)

2nd [0]: 'c'