# 题目

给你一个字符串 paragraph 和一个表示禁用词的字符串数组 banned ，返回出现频率最高的非禁用词。题目数据 保证 至少存在一个非禁用词，且答案 唯一 。

paragraph 中的单词 不区分大小写 ，答案应以 小写 形式返回。

示例 1：

输入：paragraph = "Bob hit a ball, the hit BALL flew far after it was hit.", banned = ["hit"]

输出："ball"

解释：

"hit" 出现了 3 次，但它是禁用词。

"ball" 出现了两次（没有其他单词出现这么多次），因此它是段落中出现频率最高的非禁用词。

请注意，段落中的单词不区分大小写，

标点符号会被忽略（即使它们紧挨着单词，如 "ball,"），

并且尽管 "hit" 出现的次数更多，但它不能作为答案，因为它是禁用词。

示例 2：

输入：paragraph = "a.", banned = []

输出："a"

提示：

1 <= paragraph.length <= 1000

paragraph 由英文字母、空格 ' '、和以下符号组成："!?',;."

0 <= banned.length <= 100

1 <= banned[i].length <= 10

banned[i] 仅由小写英文字母组成

# 分析

## 方法一：两次遍历

class Solution {

public:

string mostCommonWord(string paragraph, vector<string>& banned) {

unordered\_map<string, bool> bannedMap;

for (string& bannedWord : banned) {

bannedMap[bannedWord] = true;

}

unordered\_map<string, int> freqMap;

string word = "";

string mostCommonWord = "";

int maxFreq = 0;

for (char& c : paragraph) {

if (isalpha(c)) {

word += tolower(c);

} else if (!word.empty()) {

if (bannedMap.find(word) == bannedMap.end()) {

freqMap[word]++;

if (freqMap[word] > maxFreq) {

maxFreq = freqMap[word];

mostCommonWord = word;

}

}

word = "";

}

}

// 处理最后一个单词（或只有一个单词的情况）

if (!word.empty()) {

if (bannedMap.find(word) == bannedMap.end()) {

freqMap[word]++;

if (freqMap[word] > maxFreq) {

mostCommonWord = word;

}

}

}

// 处理freqMap中的所有单词

for (auto& pair : freqMap) {

if (pair.second > maxFreq) {

mostCommonWord = pair.first;

maxFreq = pair.second;

}

}

return mostCommonWord;

}

};