# 题目

给你一个字符串数组words ，找出并返回length(words[i]) \* length(words[j]) 的最大值，并且这两个单词不含有公共字母。如果不存在这样的两个单词，返回0。

示例 1：

输入：words = ["abcw","baz","foo","bar","xtfn","abcdef"]

输出：16

解释：这两个单词为 "abcw", "xtfn"。

示例 2：

输入：words = ["a","ab","abc","d","cd","bcd","abcd"]

输出：4

解释：这两个单词为 "ab", "cd"。

示例 3：

输入：words = ["a","aa","aaa","aaaa"]

输出：0

解释：不存在这样的两个单词。

提示：

2 <= words.length <= 1000

1 <= words[i].length <= 1000

words[i] 仅包含小写字母

# 分析

class Solution {

public:

int maxProduct(vector<string>& words) {

int n = words.size();

int mask[n];

memset(mask, 0, sizeof(mask));

int ans = 0;

for (int i = 0; i < n; i++) {

for (auto c : words[i]) {

// 读取每个字符并且转换为对应的数值

mask[i] |= 1 << (c - 'a');

}

for (int j = 0; j < i; j++) {

// 这两个数组的字符必须是不同的

if ((mask[i] & mask[j]) == 0) {

ans = max(ans, (int)(words[i].size() \* words[j].size()));

}

}

}

return ans;

}

};