Problem 318: Maximum Product of Word Lengths

Problem Information

Difficulty: Medium
Acceptance Rate: 60.86%
Paid Only: No
Tags: Array, String, Bit Manipulation

Problem Description

Given a string array `words`, return _the maximum value of_ `length(word[i]) * length(word[j])` _where the two words do not share common letters_. If no such two words exist, return `0`.

Example 1:

Input: words = ["abcw","baz","foo","bar","xtfn","abcdef"] **Output:** 16 **Explanation:** The two words can be "abcw", "xtfn".

Example 2:

Input: words = ["a","ab","abc","d","cd","bcd","abcd"] **Output:** 4 **Explanation:** The two words can be "ab", "cd".

Example 3:

Input: words = ["a","aa","aaa","aaaa"] **Output:** 0 **Explanation:** No such pair of words.

Constraints:

* `2 <= words.length <= 1000` * `1 <= words[i].length <= 1000` * `words[i]` consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int maxProduct(vector<string>& words) {

}
};
```

Java:

```
class Solution {
public int maxProduct(String[] words) {

}
}
```

Python3:

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