

Problem 3696: Maximum Distance Between Unequal Words in Array I

Problem Information

Difficulty: Easy

Acceptance Rate: 83.35%

Paid Only: Yes

Tags: Array, String

Problem Description

You are given a string array `words`.

Find the **maximum distance** between two **distinct** indices `i` and `j` such that:

* `words[i] != words[j]`, and * the distance is defined as `j - i + 1`.

Return the maximum distance among all such pairs. If no valid pair exists, return 0.

Example 1:

Input: words = ["leetcode", "leetcode", "codeforces"]

Output: 3

Explanation:

In this example, `words[0]` and `words[2]` are not equal, and they have the maximum distance `2 - 0 + 1 = 3`.

Example 2:

Input: words = ["a", "b", "c", "a", "a"]

Output: 4

****Explanation:****

In this example `words[1]` and `words[4]` have the largest distance of `4 - 1 + 1 = 4`.

****Example 3:****

****Input:**** words = ["z", "z", "z"]

****Output:**** 0

****Explanation:****

In this example all the words are equal, thus the answer is 0.

****Constraints:****

* `1 <= words.length <= 100` * `1 <= words[i].length <= 10` * `words[i]` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
    int maxDistance(vector<string>& words) {
        }
    };
}
```

Java:

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

Python3:

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