Problem 3265: Count Almost Equal Pairs I

Problem Information

Difficulty: Medium
Acceptance Rate: 37.96%
Paid Only: No
Tags: Array, Hash Table, Sorting, Counting, Enumeration

Problem Description

You are given an array `nums` consisting of positive integers.

We call two integers `x` and `y` in this problem **almost equal** if both integers can become equal after performing the following operation **at most once** :

* Choose **either** `x` or `y` and swap any two digits within the chosen number.

Return the number of indices `i` and `j` in `nums` where `i < j` such that `nums[i]` and `nums[j]` are **almost equal**.

Note that it is allowed for an integer to have leading zeros after performing an operation.

Example 1:

Input: nums = [3,12,30,17,21]

Output: 2

Explanation:

The almost equal pairs of elements are:

* 3 and 30. By swapping 3 and 0 in 30, you get 3. * 12 and 21. By swapping 1 and 2 in 12, you get 21.

Example 2:

Input: nums = [1,1,1,1,1]

Output: 10

Explanation:

Every two elements in the array are almost equal.

Example 3:

Input: nums = [123,231]

Output: 0

Explanation:

We cannot swap any two digits of 123 or 231 to reach the other.

Constraints:

* `2 <= nums.length <= 100` * `1 <= nums[i] <= 106`

Code Snippets

C++:

```
class Solution {
public:
int countPairs(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int countPairs(int[] nums) {



}
}
```

Python3:

```
class Solution:
def countPairs(self, nums: List[int]) -> int:
```