Problem 3267: Count Almost Equal Pairs II

Problem Information

Difficulty: Hard
Acceptance Rate: 26.79%
Paid Only: No
Tags: Array, Hash Table, Sorting, Counting, Enumeration

Problem Description

Attention : In this version, the number of operations that can be performed, has been increased to **twice**.

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

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

* 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 = [1023,2310,2130,213]

Output: 4

Explanation:

The almost equal pairs of elements are:

* 1023 and 2310. By swapping the digits 1 and 2, and then the digits 0 and 3 in 1023, you get 2310. * 1023 and 213. By swapping the digits 1 and 0, and then the digits 1 and 2 in 1023, you get 0213, which is 213. * 2310 and 213. By swapping the digits 2 and 0, and then the digits 3 and 2 in 2310, you get 0213, which is 213. * 2310 and 2130. By swapping the digits 3 and 1 in 2310, you get 2130.

Example 2:

Input: nums = [1,10,100]

Output: 3

Explanation:

The almost equal pairs of elements are:

* 1 and 10. By swapping the digits 1 and 0 in 10, you get 01 which is 1. * 1 and 100. By swapping the second 0 with the digit 1 in 100, you get 001, which is 1. * 10 and 100. By swapping the first 0 with the digit 1 in 100, you get 010, which is 10.

Constraints:

* `2 <= nums.length <= 5000` * `1 <= nums[i] < 107`

Code Snippets

C++:

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


    }
};
```

Java:

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

```
    }
}
```

Python3:

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