Problem 3551: Minimum Swaps to Sort by Digit Sum

Problem Information

Difficulty: Medium
Acceptance Rate: 50.29%
Paid Only: No
Tags: Array, Hash Table, Sorting

Problem Description

You are given an array `nums` of **distinct** positive integers. You need to sort the array in **increasing** order based on the sum of the digits of each number. If two numbers have the same digit sum, the **smaller** number appears first in the sorted order.

Return the **minimum** number of swaps required to rearrange `nums` into this sorted order.

A **swap** is defined as exchanging the values at two distinct positions in the array.

Example 1:

Input: nums = [37,100]

Output: 1

Explanation:

* Compute the digit sum for each integer: `[3 + 7 = 10, 1 + 0 + 0 = 1] -> [10, 1]` * Sort the integers based on digit sum: `[100, 37]`. Swap `37` with `100` to obtain the sorted order. * Thus, the minimum number of swaps required to rearrange `nums` is 1.

Example 2:

Input: nums = [22,14,33,7]

Output: 0

Explanation:

* Compute the digit sum for each integer: `[2 + 2 = 4, 1 + 4 = 5, 3 + 3 = 6, 7 = 7] -> [4, 5, 6, 7]` * Sort the integers based on digit sum: `[22, 14, 33, 7]`. The array is already sorted. * Thus, the minimum number of swaps required to rearrange `nums` is 0.

Example 3:

Input: nums = [18,43,34,16]

Output: 2

Explanation:

* Compute the digit sum for each integer: `[1 + 8 = 9, 4 + 3 = 7, 3 + 4 = 7, 1 + 6 = 7] -> [9, 7, 7, 7]` * Sort the integers based on digit sum: `[16, 34, 43, 18]`. Swap `18` with `16`, and swap `43` with `34` to obtain the sorted order. * Thus, the minimum number of swaps required to rearrange `nums` is 2.

Constraints:

* `1 <= nums.length <= 105` * `1 <= nums[i] <= 109` * `nums` consists of **distinct** positive integers.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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