Problem 3079: Find the Sum of Encrypted Integers

Problem Information

Difficulty: Easy
Acceptance Rate: 74.41%
Paid Only: No
Tags: Array, Math

Problem Description

You are given an integer array `nums` containing **positive** integers. We define a function `encrypt` such that `encrypt(x)` replaces **every** digit in `x` with the **largest** digit in `x`. For example, `encrypt(523) = 555` and `encrypt(213) = 333`.

Return _the**sum** of encrypted elements_.

Example 1:

Input: nums = [1,2,3]

Output: 6

Explanation: The encrypted elements are `[1,2,3]`. The sum of encrypted elements is `1 + 2 + 3 == 6`.

Example 2:

Input: nums = [10,21,31]

Output: 66

Explanation: The encrypted elements are `[11,22,33]`. The sum of encrypted elements is `11 + 22 + 33 == 66`.

Constraints:

* `1 <= nums.length <= 50` * `1 <= nums[i] <= 1000`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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