Problem 2221: Find Triangular Sum of an Array

Problem Information

Difficulty: Medium
Acceptance Rate: 81.97%
Paid Only: No
Tags: Array, Math, Simulation, Combinatorics

Problem Description

You are given a **0-indexed** integer array `nums`, where `nums[i]` is a digit between `0` and `9` (**inclusive**).

The **triangular sum** of `nums` is the value of the only element present in `nums` after the following process terminates:

1. Let `nums` comprise of `n` elements. If `n == 1`, **end** the process. Otherwise, **create** a new **0-indexed** integer array `newNums` of length `n - 1`. 2. For each index `i`, where `0 <= i < n - 1`, **assign** the value of `newNums[i]` as `(nums[i] + nums[i+1]) % 10`, where `%` denotes modulo operator. 3. **Replace** the array `nums` with `newNums`. 4. **Repeat** the entire process starting from step 1.

Return _the triangular sum of_ `nums`.

Example 1:

Input: nums = [1,2,3,4,5] **Output:** 8 **Explanation:** The above diagram depicts the process from which we obtain the triangular sum of the array.

Example 2:

Input: nums = [5] **Output:** 5 **Explanation:** Since there is only one element in nums, the triangular sum is the value of that element itself.

Constraints:

* `1 <= nums.length <= 1000` * `0 <= nums[i] <= 9`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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