Problem 1313: Decompress Run-Length Encoded List

Problem Information

Difficulty: Easy
Acceptance Rate: 86.13%
Paid Only: No
Tags: Array

Problem Description

We are given a list `nums` of integers representing a list compressed with run-length encoding.

Consider each adjacent pair of elements `[freq, val] = [nums[2*i], nums[2*i+1]]` (with `i >= 0`). For each such pair, there are `freq` elements with value `val` concatenated in a sublist. Concatenate all the sublists from left to right to generate the decompressed list.

Return the decompressed list.

Example 1:

Input: nums = [1,2,3,4] **Output:** [2,4,4,4] **Explanation:** The first pair [1,2] means we have freq = 1 and val = 2 so we generate the array [2]. The second pair [3,4] means we have freq = 3 and val = 4 so we generate [4,4,4]. At the end the concatenation [2] + [4,4,4] is [2,4,4,4].

Example 2:

Input: nums = [1,1,2,3] **Output:** [1,3,3]

Constraints:

* `2 <= nums.length <= 100` * `nums.length % 2 == 0` * `1 <= nums[i] <= 100`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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