1313. Decompress Run-Length Encoded List

Easy 116 378 Add to List Share

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]
```

C++ Autocomplete

```cpp
class Solution {
public:
    vector<int> decompressRLElist(vector<int>& nums) {
        vector<int> ans;
        int n=nums.size();
        for(int i=0;i<n;i+=2){
            int f=nums[i];
            int v=nums[i+1];
            for(int j=0;j<f;j++){
                ans.push_back(v);
            }
        }
        return ans;

    }
};
```

Testcase Run Code Result Debugger

Accepted Runtime: 4 ms

Your input [1,2,3,4]

Output [2,4,4,4] Diff

Expected [2,4,4,4]