

1313. Decompress Run-Length Encoded List

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 $i \geq 0$). 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.

Solution:

```
class Solution(object):
```

```
    def decompressRLElist(self, nums):
```

```
        """
```

```
        :type nums: List[int]
```

```
        :rtype: List[int]
```

```
        """
```

```
        new_list = list()
```

```
        for i in range(0, len(nums)-1, 2):
```

```
            for j in range(0, nums[i]):
```

```
                new_list.append(nums[i+1])
```

```
        return new_list
```

Description

Solution

Submissions

Discuss (588)

Python

Autocomplete

Success

Details

Runtime: 68 ms, faster than 15.04% of Python online submissions for Decompress Run-Length Encoded List.

Memory Usage: 12.8 MB, less than 59.65% of Python online submissions for Decompress Run-Length Encoded List.

Next challenges:

String Compression

Show off your acceptance:

f

t

in

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	68 ms	12.8 MB	python

1

2

3

4

5

6

7

8

9

10

11

12

13

```
class Solution(object):
    def decompressRLElist(self, nums):
        """
        :type nums: List[int]
        :rtype: List[int]
        """
        new_list = list()
        for i in range(0, len(nums)-1, 2):
            for j in range(0, nums[i]):
                new_list.append(nums[i+1])
        return new_list
```

Testcase

Run Code Result

Debugger

Accepted

Runtime: 20 ms

Your input

[1,2,3,4]

Output

[2,4,4,4]

Expected

[2,4,4,4]