Leetcode 2149.
Rearrange Array Elements by Sign

2024.02.18

원루빈

* Problem Description - Rearrange Array Elements by Sign

You are given a **0-indexed** integer array nums of even length consisting of an **equal** number of positive and negative integers.

You should return the array of **nums** such that the the array follows the given conditions:

[1] Every consecutive pair of integers have opposite signs.
[2] For all integers with the same sign, the order in which they were present in nums is preserved.
[3] The rearranged array begins with a positive integer.

Return the modified array after rearranging the elements to satisfy the aforementioned conditions.

Example:

Input: nums = [3,1,-2,-5,2,-4]
Output: [3,-2,1,-5,2,-4]

* Stable Sorting

- Rearrangement based on sign while preserving original order

[DEF] stable sorting: if two elements have the same key, the one that appeared earlier in the input will also appear in the sorted output.

E.g.

list=[1, 7(1), 3, 5, 4, 7(2), 9]

Stable sorting: list=[1, 3, 4, 5, 7(1), 7(2), 9]
Unstable sorting: list=[1, 3, 4, 5, 7(2), 7(1), 9]

[2] For all integers with the same sign, the order in which they were present in nums is preserved.

1 [Approach 1] Separate positive and negative numbers

1. Separate positive and negative numbers (unstable sorting method)

```
positives = [i for i in nums if i > 0]
negatives = [i for i in nums if i < 0]
```

2. Merge positive and negative numbers in alternating order

```
for pos, neg in zip(positives, negatives):
    result.append(pos)
    result.append(neg)
```

1 [Approach 1] Solution

```python
class Solution:
    def rearrangeArray_1(self, nums):
        # Runtime: Beats 97.89% of users with Python3
        # Memory: Beats 48.57% of users with Python3
        # Separate positive and negative numbers
        positives = [i for i in nums if i > 0]
        negatives = [i for i in nums if i < 0]

        result = []

        # Merge positive and negative numbers in alternating order, starting with pos
        for pos, neg in zip(positives, negatives):
            result.append(pos)
            result.append(neg)
        return result
```

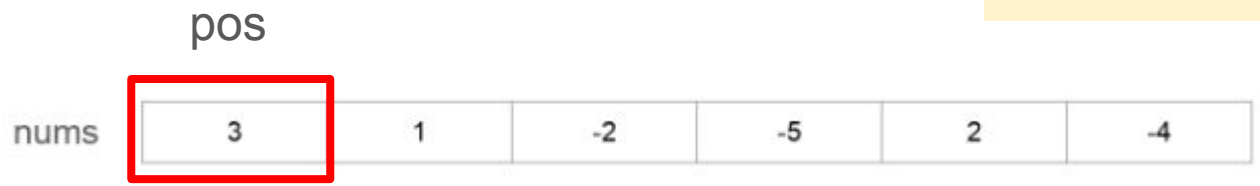
2 [Approach 2] Two pointers approach

pos_pt = 0
neg_pt = 1

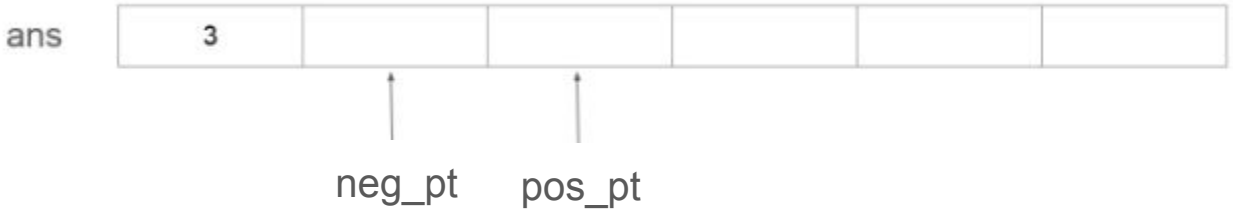
ans = [0] * len(nums)

2 [Approach 2] Two pointers approach

pos_pt = 0 + 2 = 2
neg_pt = 1

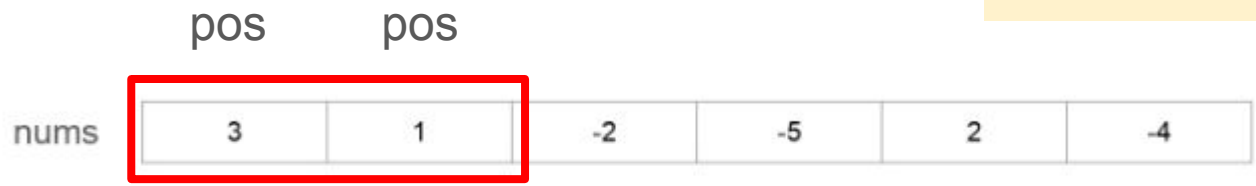


ans = [0] * len(nums)

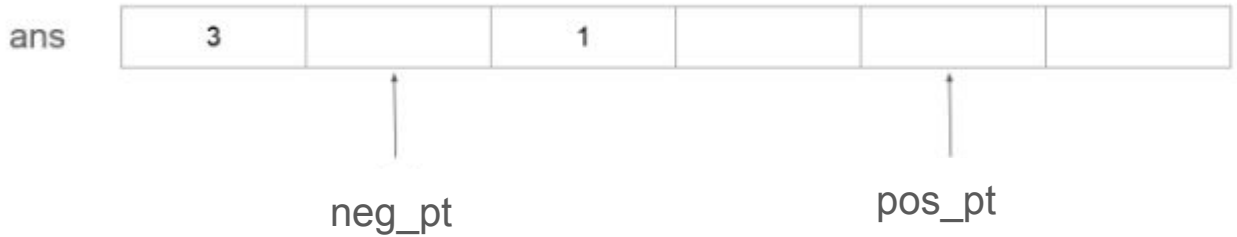


2 [Approach 2] Two pointers approach

pos_pt = 2 + 2 = 4
neg_pt = 1



ans = [0] * len(nums)



2 [Approach 2] Two pointers approach

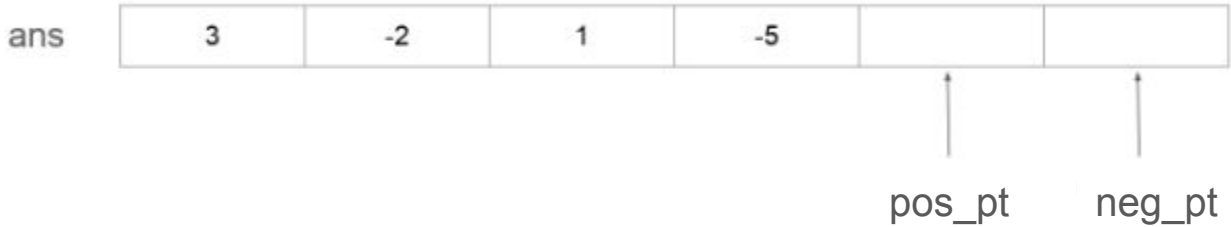
pos_pt = 4
neg_pt = 1 + 2 = 3

ans = [0] * len(nums)

2 [Approach 2] Two pointers approach

pos_pt = 4
neg_pt = 3 + 2 = 5

ans = [0] * len(nums)



2 [Approach 2] Two pointers approach

pos_pt = Done
neg_pt = 5

ans = [0] * len(nums)

2 [Approach 2] Two pointers approach

pos_pt = Done
neg_pt = Done

ans = [0] * len(nums)

2 Approach 2 Solution (2 pointers)

```python
def rearrangeArray_2_fixed(self, nums): # Two pointers approach modified
    # Runtime: Beats 89.67% of users with Python3
    # Memory: Beats 65.62% of users with Python3
    ans = [0] * len(nums)
    pos_pt, neg_pt = 0, 1  # Start pointers for positive and negative numbers

    for num in nums:
        if num > 0:
            ans[pos_pt] = num
            pos_pt += 2
        else:
            ans[neg_pt] = num
            neg_pt += 2

    return ans
```

감사합니다!

THANK YOU