2149. Rearrange Array Elements by Sign

Solved

Medium | Topics | Companies | Hint

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

You should **rearrange** the elements of `nums` such that the modified 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 1:

> **Input:** nums = [3,1,-2,-5,2,-4]
> **Output:** [3,-2,1,-5,2,-4]
> **Explanation:**
> The positive integers in nums are [3,1,2]. The negative integers are [-2,-5,-4].
> The only possible way to rearrange them such that they satisfy all conditions is [3,-2,1,-5,2,-4].
> Other ways such as [1,-2,2,-5,3,-4], [3,1,2,-2,-5,-4], [-2,3,-5,1,-4,2] are incorrect because they do not satisfy one or more conditions.

Example 2:

> **Input:** nums = [-1,1]
> **Output:** [1,-1]
> **Explanation:**
> 1 is the only positive integer and -1 the only negative integer in nums.
> So nums is rearranged to [1,-1].

Constraints:

- `2 <= nums.length <= 2 * 10^5`
- `nums.length` is **even**
- `1 <= |nums[i]| <= 10^5`
- `nums` consists of **equal** number of positive and negative integers.

Approach 1: Using two xtra n/2 sized arrays.

→ Parray and Narray
→ Traverse input array filling parray and narray.
→ Now modify given input array

→ How reading given input array:

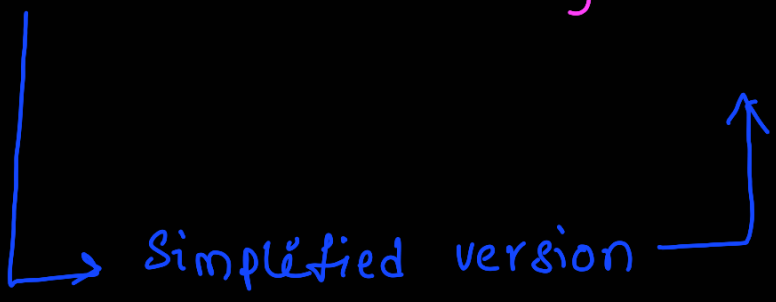
pindex = 0

nindex = 1

```
for(i : 0 to n-1)
{
    if(i is even)
        nums[i] = parray[pindex++]
    else
        nums[i] = narray[nindex++]
}
```

(or)

```
for(i : 0 to parray.size() - 1)
{
    nums[2*i] = parray[i]
    nums[2*i+1] = narray[i]
}
```



T(n) : O(n) + O(n)

S(n) : O(n/2) + O(n/2)

Approach 2: Using two pointers and in one pass

→ vector<int> ans(n)

→ pindex = 0 , nindex = 0

```
for(i : 0 to n-1)
{
    if(nums[i] is positive)
    {   ans[pindex] = nums[i]
        pindex = pindex + 2
    }
    else
    {   ans[nindex] = nums[i]
        nindex = nindex + 2
    }
```

}

T(n) : O(n)
S(n) : O(n)

follow up question: what if no.of positives is not equal no.of negatives and in this case we should follow the rule for same no.of elements and then remaining excess elements must be appended at the end of list in the same order. how do we approach this?

To implement this we need to go for approach 1 and approach 2 won't work bcoz i will go out of bounds.

→ Parray, narray
→ fill parray, narray
→ now
n = min(parray size, narray size)
→
for(i: 0 to n-1)
{
nums[2*i] = parray[i]
nums[2*i+1] = narray[i]
}

→ now
x = 2*n ; i = n

for (x : 2*n to nums size -1)

{

nums[x] = Parray size is more ? parray[i] : narray[i]

i++

}

T(n) : O(n) + O(min(parray size, narraysize))
+ O(n - min(parraysize, narraysize) #2)

S(n) : O(K) + O(n-K) = O(n)