

HW CLASS: 10 12/09/2023

<https://www.linkedin.com/in/manojofficialmj/>

### 0. Re-arrange array elements (Leetcode-2149)

Input	nums	<table border="1"> <tr> <td>3</td><td>1</td><td>-2</td><td>-5</td><td>2</td><td>-4</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	3	1	-2	-5	2	-4	0	1	2	3	4	5	$\text{size} = 6$
3	1	-2	-5	2	-4										
0	1	2	3	4	5										

Output	<table border="1"> <tr> <td>3</td><td>-2</td><td>1</td><td>-5</td><td>2</td><td>-4</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	3	-2	1	-5	2	-4	0	1	2	3	4	5
3	-2	1	-5	2	-4								
0	1	2	3	4	5								

Observation

ANS	<table border="1"> <tr> <td>+ve</td><td>-ve</td><td>+ve</td><td>-ve</td><td>+ve</td><td>-ve</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	+ve	-ve	+ve	-ve	+ve	-ve	0	1	2	3	4	5
+ve	-ve	+ve	-ve	+ve	-ve								
0	1	2	3	4	5								

Note

- ① Array's size always even
- ② Negative no's == positive no's
- ③ Pairs of number always start from positive no.

DryRun

iteration: 0	nums	<table border="1"> <tr> <td>3</td><td>1</td><td>-2</td><td>-5</td><td>2</td><td>-4</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	3	1	-2	-5	2	-4	0	1	2	3	4	5
3	1	-2	-5	2	-4									
0	1	2	3	4	5									

$\text{index} = 0$

$\text{posIndex} = 0$

$\text{nugIndex} = 1$

①  $\text{nums}[\text{index}] > 0$

$$3 > 0$$

$\text{ans}[\text{posIndex}] = \text{nums}[\text{index}]$

$$\text{ans}[0] = 3$$

$$\text{posIndex} = \text{posIndex} + 2$$

$$\begin{aligned} &= 0 + 2 \\ &= 2 \end{aligned}$$

ans	<table border="1"> <tr> <td>3</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	3	0	0	0	0	0	0	1	2	3	4	5
3	0	0	0	0	0								
0	1	2	3	4	5								

iteration: 1

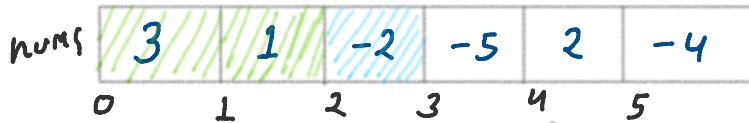
nums	<table border="1"> <tr> <td>3</td><td>1</td><td>-2</td><td>-5</td><td>2</td><td>-4</td></tr> <tr> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr> </table>	3	1	-2	-5	2	-4	0	1	2	3	4	5
3	1	-2	-5	2	-4								
0	1	2	3	4	5								

$\text{index} = 1$   
 $\text{posIndex} = 2$   
 $\text{nugIndex} = 1$

①  $\text{nums}[\text{index}] > 0$   
 $1 > 0$   
 $\text{ans}[\text{posIndex}] = \text{nums}[\text{index}];$   
 $\text{ans}[2] = 1$   
 $\text{posIndex} = \text{posIndex} + 2;$   
 $= 2 + 2$   
 $= 4$



Iteration: 2

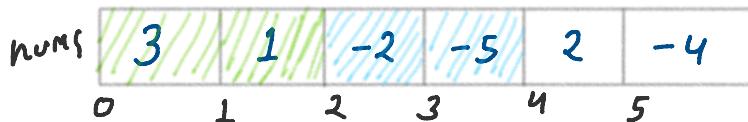


$\text{index} = 2$   
 $\text{posIndex} = 4$   
 $\text{nugIndex} = 1$

①  $\text{nums}[\text{index}] < 0$   
 $-2 < 0$   
 $\text{ans}[\text{nugIndex}] = \text{nums}[\text{index}];$   
 $\text{ans}[1] = -2$   
 $\text{nugIndex} = \text{nugIndex} + 2;$   
 $= 1 + 2$   
 $= 3$



Iteration: 3

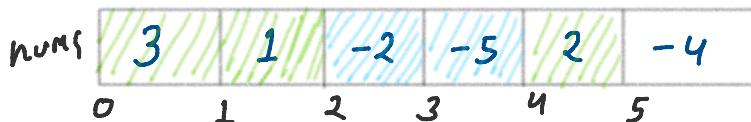


$\text{index} = 3$   
 $\text{posIndex} = 4$   
 $\text{nugIndex} = 3$

①  $\text{nums}[\text{index}] < 0$   
 $-5 < 0$   
 $\text{ans}[\text{nugIndex}] = \text{nums}[\text{index}];$   
 $\text{ans}[3] = -5$   
 $\text{nugIndex} = \text{nugIndex} + 2;$   
 $= 3 + 2$   
 $= 5$



Iteration: 4



$\text{index} = 4$   
 $\text{posIndex} = 4$   
 $\text{nugIndex} = 5$

①  $\text{nums}[\text{index}] > 0$   
 $2 > 0$   
 $\text{ans}[\text{posIndex}] = \text{nums}[\text{index}];$   
 $\text{ans}[4] = 2$   
 $\text{posIndex} = \text{posIndex} + 2;$   
 $= 4 + 2$



0

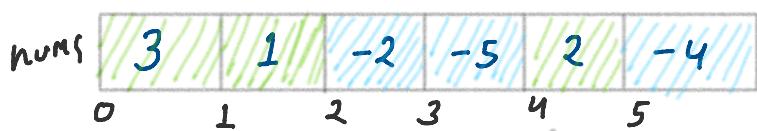
$$\text{posIndex} = \text{posIndex} + 1$$

$$= 4 + 2$$

$$= 6$$



Iteration: 5



$$\text{index} = 5$$

$$\text{posIndex} = 6$$

$$\text{negIndex} = 5$$

①  $\text{nums}[\text{index}] < 0$

$$-4 < 0$$

$$\text{ans}[\text{negIndex}] = \text{nums}[\text{index}];$$

$$\text{ans}[5] = -4$$

$$\text{negIndex} = \text{negIndex} + 2;$$

$$= 5 + 2$$

$$= 7$$



Iteration: 6

$(\text{index} < \text{size})$   
 $6 < 6$  X END

Final Output

```
// HW 00: Re-arrange array elements (Leetcode-2149)

class Solution {
public:
    vector<int> rearrangeArray(vector<int>& nums) {
        // Size of nums
        int size=nums.size();

        // Create new vector array
        vector<int> ans(size,0);

        // Create positive index and negative index
        int posIndex=0;
        int negIndex=1;

        // Iterate the entire array
        for(int index=0;index<size;index++){
            // When element is positive then use even index in ans
            if(nums[index]>0){
                ans[posIndex]=nums[index];
                posIndex+=2;
            }
            // When element is positive then use even index in ans
            else{
                ans[negIndex]=nums[index];
                negIndex+=2;
            }
        }

        // Return new array
        return ans;
    }
};
```

T.C.  $\Rightarrow O(N)$