

**R15**  
**Array questions (Intermediate)**

# 1480. Running Sum of 1d Array

Solved ✓

Easy

Topics

Companies

Hint

Given an array `nums`. We define a running sum of an array as `runningSum[i] = sum(nums[0]...nums[i])`.

Return the running sum of `nums`.

↓ ↓ ↓ ↓  
1, 2, 3, 4  
  
1 3 6 10

## Example 1:

Input: `nums = [1,2,3,4]`

Output: `[1,3,6,10]`

Explanation: Running sum is obtained as follows: `[1, 1+2, 1+2+3, 1+2+3+4]`.

## Example 2:

Input: `nums = [1,1,1,1,1]`

Output: `[1,2,3,4,5]`

Explanation: Running sum is obtained as follows: `[1, 1+1, 1+1+1, 1+1+1+1, 1+1+1+1+1]`.

## Example 3:

Input: `nums = [3,1,2,10,1]`

Output: `[3,4,6,16,17]`

↔ → 3 4 6 16 17

prefix

1	2	3	4	5
---	---	---	---	---

↓ ↓ ↓ ↓ ↓

1 3 6 10 15

1	2	3	4	5
---	---	---	---	---



15	14	12	9	5
----	----	----	---	---

suffix[i]  
= sum

sum = 0

suffix



Logic



SC  $\rightarrow O(n)$

1	2	3	4	5	6
---	---	---	---	---	---



sum = sum + an[i]

sum = 1  
↑ 2  
= 3

1	3	6	10	15	21
---	---	---	----	----	----

prefix[i] = sum

TC  $\rightarrow O(n)$

# 303. Range Sum Query - Immutable

Solved

Easy

Topics

Companies

Given an integer array `nums`, handle multiple queries of the following type:

1. Calculate the **sum** of the elements of `nums` between indices `left` and `right` **inclusive** where `left <= right`.

Implement the `NumArray` class:

- `NumArray(int[] nums)` Initializes the object with the integer array `nums`.
- `int sumRange(int left, int right)` Returns the **sum** of the elements of `nums` between indices `left` and `right` **inclusive** (i.e. `nums[left] + nums[left + 1] + ... + nums[right]`).

Example 1:

Input

```
["NumArray", "sumRange", "sumRange", "sumRange"]  
[[-2, 0, 3, -5, 2, -1]], [0, 2], [2, 5], [0, 5]]
```

Output

```
[null, 1, -1, -3]
```

Explanation

```
NumArray numArray = new NumArray([-2, 0, 3, -5, 2, -1]);  
numArray.sumRange(0, 2); // return (-2) + 0 + 3 = 1  
numArray.sumRange(2, 5); // return 3 + (-5) + 2 + (-1) = -1  
numArray.sumRange(0, 5); // return (-2) + 0 + 3 + (-5) + 2 + (-1) =  
-3
```

0	1	2	3	4
1	2	3	4	5
1	3	6	10	15

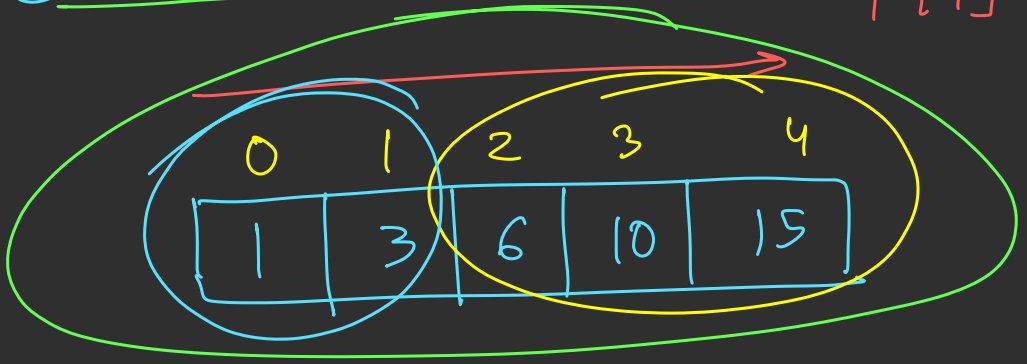
0 - 3  
1 - 4

✓  $N \rightarrow 10^5$

$$p[2-1] = p[1]$$

$$\textcircled{2} \dots 4$$

$$p[4]$$



$p[i] \rightarrow i^{\text{th}}$  element sum

$$p[4] - p[1] \rightarrow \textcircled{12}$$

$$1^0 \quad 2^1 \quad 3^2 \quad 4^3 \quad 5^4$$

└──────────────────┘

└────────┘  $\textcircled{12}$

Consider an array  $A[]$  of integers and following two types of queries.

1.  $\text{update}(l, r, x)$  : Adds  $x$  to all values from  $A[l]$  to  $A[r]$  (both inclusive).
2.  $\text{printArray}()$  : Prints the current modified array.

Examples :

Input :  $A [] \{ 10, 5, 20, 40 \}$

$\text{update}(0, 1, 10)$

$\text{printArray}()$

$\text{update}(1, 3, 20)$

$\text{update}(2, 2, 30)$

$\text{printArray}()$

Output : 20 15 20 40

20 35 70 60

Explanation : The query  $\text{update}(0, 1, 10)$

adds 10 to  $A[0]$  and  $A[1]$ . After update,

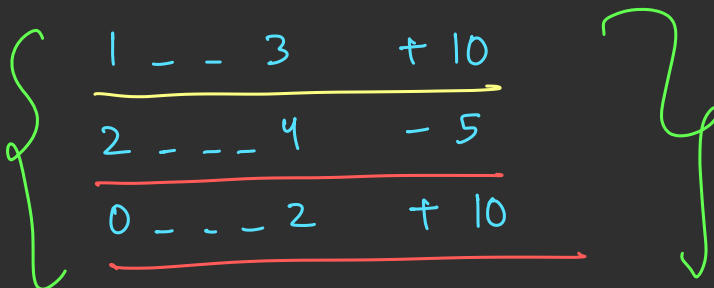
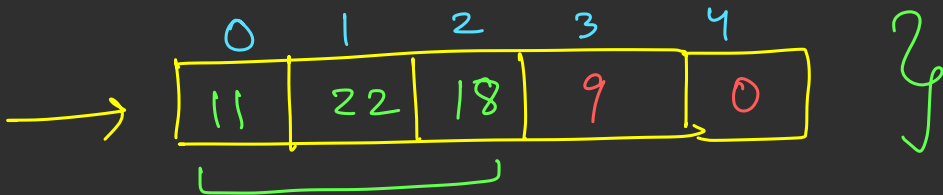
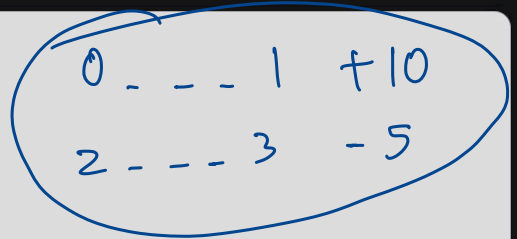
$A[]$  becomes  $\{20, 15, 20, 40\}$

Query  $\text{update}(1, 3, 20)$  adds 20 to  $A[1]$ ,

$A[2]$  and  $A[3]$ . After update,  $A[]$  becomes  $\{20, 35, 40, 60\}$ .

Query  $\text{update}(2, 2, 30)$  adds 30 to  $A[2]$ .

After update,  $A[]$  becomes  $\{20, 35, 70, 60\}$ .



diff

L →

10	10	-5	-10	-10	5
0	1	2	3	4	5

Difference array

0 - - 2 +10

10	20	15	5	-5	0
----	----	----	---	----	---

sum 20 ~~+10~~ ~~+10~~  
~~-5~~ ~~-10~~  
~~-10~~ ~~+5~~

l - - r

→ diff[l] += val  
→ diff[r+1] -= val

↓  
n<sup>th</sup>

1	2	3	4	5
---	---	---	---	---

→

11	22	18	9	0
----	----	----	---	---

←



$$\frac{N}{N+1}$$

↑

$$N = 6$$

→

$$\frac{n-1+1}{n} \rightarrow \frac{n}{n}$$

$$\begin{matrix} \leftarrow & 10 & 10 & \rightarrow \\ \downarrow & & & \end{matrix}$$

$$X \rightarrow 0$$

0	10	-5	0	-10	5
0	1	2	3	4	5



$$\begin{matrix} \leftarrow & 5 & 5 & \rightarrow \\ \leftarrow & & & \rightarrow \end{matrix}$$

$$\begin{matrix} 1 & - & - & 3 & +10 \\ 2 & - & - & 4 & -5 \end{matrix}$$

0	10	5	5	-5	0
---	----	---	---	----	---

# 75. Sort Colors

Solved ✓

Medium

Topics

Companies

Hint

Given an array `nums` with `n` objects colored red, white, or blue, sort them **in-place** so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers `0`, `1`, and `2` to represent the color red, white, and blue, respectively.

You must solve this problem without using the library's sort function.

Example 1:

Input: `nums = [2,0,2,1,1,0]`

Output: `[0,0,1,1,2,2]`

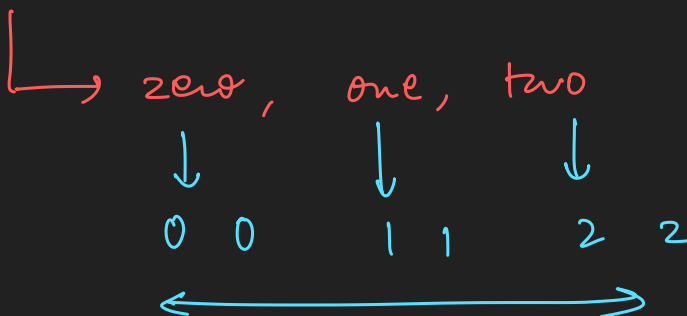
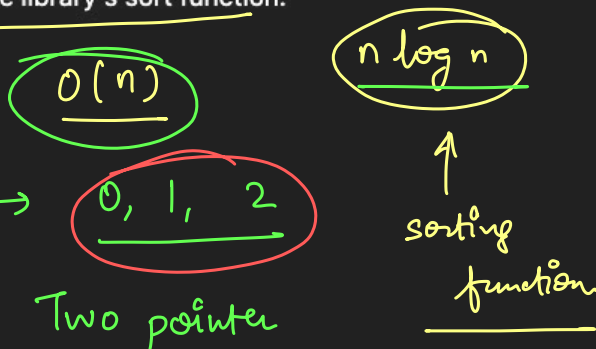
Example 2:

Input: `nums = [2,0,1]`

Output: `[0,1,2]`

Constraints:

- `n == nums.length`
- `1 <= n <= 300`
- `nums[i]` is either `0`, `1`, or `2`.



0 0 1 1 2 2  
~~0~~ ~~1~~ ~~2~~ ~~0~~ ~~1~~ ~~2~~

z = 2      t = 2  
0 2 2

$O(1)$

↑

SC

~~0~~ ~~2~~ ~~2~~ ~~2~~ ~~1~~ ~~0~~ ~~0~~  
0 0 0 1 2 2 2

$z = 3$

$0 \rightarrow z = 1$

$t = 3$

TC  $\rightarrow O(n)$

↑  
1  
1

↑

zero's, one's, two's

→ Update  $\rightarrow O(n)$

$z \rightarrow 0$

↓

$O(n) \rightarrow TC$

$O(1) \rightarrow SC$

$N - (z + 0)$

↑

two's  $\rightarrow$  2's

# 189. Rotate Array

Solved ✓

Medium

Topics

Companies

Hint

Given an integer array `nums`, rotate the array to the right by `k` steps, where `k` is non-negative.

Example 1:

Input: `nums = [1,2,3,4,5,6,7]`

`k = 3`

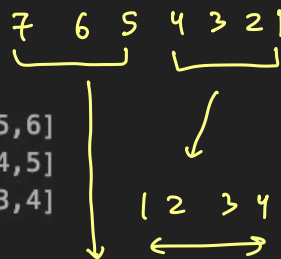
Output: `[5,6,7,1,2,3,4]`

Explanation:

rotate 1 steps to the right: `[7,1,2,3,4,5,6]`

rotate 2 steps to the right: `[6,7,1,2,3,4,5]`

rotate 3 steps to the right: `[5,6,7,1,2,3,4]`



Example 2:

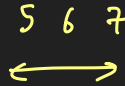
Input: `nums = [-1,-100,3,99]`, `k = 2`

Output: `[3,99,-1,-100]`

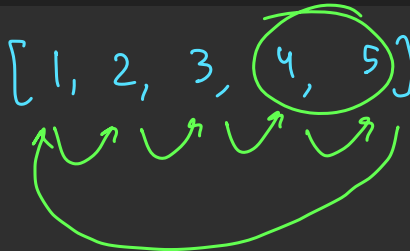
Explanation:

rotate 1 steps to the right: `[99,-1,-100,3]`

rotate 2 steps to the right: `[3,99,-1,-100]`



`[4, 5, 1, 2, 3]`



`k = 3`

`[5, 1, 2, 3, 4]`

$$k = 3, N = 3$$

↑  
size

$$\underline{1 \quad 2 \quad 3}$$

$$k = 1$$

$$\underline{3 \quad 1 \quad 2}$$

$$\underline{k = 2}$$

$$\underline{2 \quad 3 \quad 1}$$

$$\underline{k = 3}$$

$$\underline{SC \rightarrow O(n)}$$

↑

$$\underline{k = 10^9}$$

$$\textcircled{N}$$

$$k = 4 \rightarrow \textcircled{-3} \rightarrow \text{original array}$$

$$\underline{4 \% 3} \rightarrow \underline{1}$$

$$k = 7 - 3 - 3 = \textcircled{1}$$

$$\underline{7 \% 3}$$

$$k \geq N$$

$$\rightarrow \underline{k = k \% N}$$

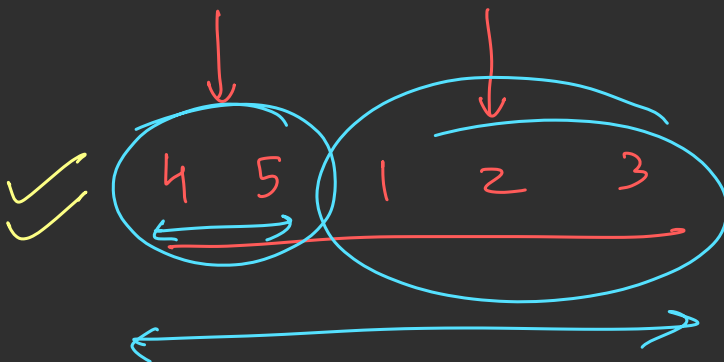
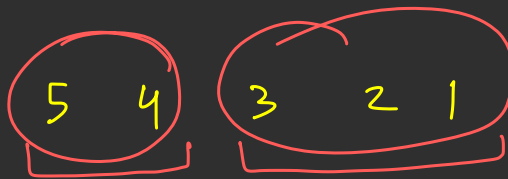
$$10 - 3 - 3 - 3 = \underline{10 \% 3} \rightarrow \textcircled{1}$$

Reverse  $\rightarrow$  Two pointer  $\rightarrow O(n)$

1 2 3 4 5

$0 - k - 1$

$k - n - 1$



SC  $\rightarrow O(1)$

$k = 2$

Reverse

Full

$O(n)$   
 $\uparrow$