

307. Range Sum Query - Mutable

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

1. Update the value of an element in nums.
2. Calculate the sum of the elements of nums between indices left and right inclusive where $\text{left} \leq \text{right}$.

Implement the NumArray class:

- NumArray(int[] nums) Initializes the object with the integer array nums.
- void update(int index, int val) Updates the value of nums[index] to be val.
- int sumRange(int left, int right) Returns the sum of the elements of nums between indices left and right inclusive (i.e. $\text{nums}[\text{left}] + \text{nums}[\text{left} + 1] + \dots + \text{nums}[\text{right}]$).

Example 1:

Input

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

Output

- [null, 9, null, 8]

Explanation

- NumArray numArray = new NumArray([1, 3, 5]);
- numArray.sumRange(0, 2); // return $1 + 3 + 5 = 9$
- numArray.update(1, 2); // nums = [1, 2, 5]
- numArray.sumRange(0, 2); // return $1 + 2 + 5 = 8$

Constraints:

- $1 \leq \text{nums.length} \leq 3 * 10^4$
- $-100 \leq \text{nums}[i] \leq 100$
- $0 \leq \text{index} < \text{nums.length}$
- $-100 \leq \text{val} \leq 100$
- $0 \leq \text{left} \leq \text{right} < \text{nums.length}$
- At most $3 * 10^4$ calls will be made to update and sumRange.