840. Range Sum Query - Mutable

* [Description](http://www.lintcode.com/en/problem/range-sum-query-mutable/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/range-sum-query-mutable/#note)
* [Testcase](http://www.lintcode.com/en/problem/range-sum-query-mutable/#testcase)
* [Judge](http://www.lintcode.com/en/problem/range-sum-query-mutable/#judge)

[Discuss](http://www.lintcode.com/discuss)

Given an integer array nums, find the sum of the elements between indices i and j (i ≤ j), inclusive.

The update(i, val) function modifies nums by updating the element at index i to val.

 Notice

1.The array is only modifiable by the update function.  
2.You may assume the number of calls to update and sumRange function is distributed evenly.

Have you met this question in a real interview?

Yes

**Example**

Given nums = [1, 3, 5]

sumRange(0, 2) -> 9

update(1, 2)

sumRange(0, 2) -> 8

[**Tags**](http://www.lintcode.com/en/problem/range-sum-query-mutable/#tags)

[**Binary Index Tree**](http://www.lintcode.com/tag/binary-index-tree/)[**Facebook**](http://www.lintcode.com/tag/facebook/)

[**Related Problems**](http://www.lintcode.com/en/problem/range-sum-query-mutable/#related)

<http://www.lintcode.com/en/problem/range-sum-query-mutable/#>

**class** NumArray {

**int**[] numeros;

**public** NumArray(**int**[] nums) {

        numeros = **new** **int**[nums.length];

        System.arraycopy(nums, 0, numeros, 0, numeros.length);

    }

**public** **void** update(**int** i, **int** val) {

        numeros[i] = val;

    }

**public**  **int** sumRange(**int** i, **int** j) {

*//HashMap<Integer, Integer> hm = new HashMap();*

**int**[] hm = **new** **int**[numeros.length];

**int** sum =0;

**for**(**int** k =0; k<numeros.length; k++) {

            sum += numeros[k];

            hm[k] = sum;

        }

**if**(i-1< 0) {

**return** hm[j];

        }

**return** hm[j] - hm[i-1];

    }

}

***/\*\****

***\* Your NumArray object will be instantiated and called as such:***

***\* NumArray obj = new NumArray(nums);***

***\* obj.update(i,val);***

***\* int param\_2 = obj.sumRange(i,j);***

***\*/***