943. Range Sum Query - Immutable

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

 Notice

1. You may assume that the array does not change.
2. There are many calls to sumRangefunction.

Have you met this question in a real interview?

Yes

**Example**

Given nums = [-2, 0, 3, -5, 2, -1]

sumRange(0, 2) -> 1

sumRange(2, 5) -> -1

sumRange(0, 5) -> -3

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

[**Dynamic Programming**](http://www.lintcode.com/tag/dynamic-programming/)[**Palantir**](http://www.lintcode.com/tag/palantir/)

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

[**Medium**Range Sum Query - Mutable**43 %**](http://www.lintcode.com/problem/range-sum-query-mutable)[**Medium**Range Sum Query 2D - Mutable**38 %**](http://www.lintcode.com/problem/range-sum-query-2d-mutable)[**Medium**Range Sum Query 2D - Immutable](http://www.lintcode.com/problem/range-sum-query-2d-immutable)

[http://www.lintcode.com/en/problem/range-sum-query-immutable/#](http://www.lintcode.com/en/problem/range-sum-query-immutable/)

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication23;

**import** java.util.HashMap;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication23 {

***/\*\****

***\* @param args the command line arguments***

***\*/***

**static** **class** NumArray {

**int**[] numeros;

**public** NumArray(**int**[] nums) {

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

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

    }

**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];

    }

   }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

**int**[] nums = {-2, 0, 3, -5, 2, -1};

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

**int** i = 0, j = 2;

**int** param\_1 = NumArray.sumRange(i, j);

        System.out.println(param\_1);

    }

}