Problem 2695: Array Wrapper

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

Create a class

ArrayWrapper

that accepts an array of integers in its constructor. This class should have two features:

When two instances of this class are added together with the

+

operator, the resulting value is the sum of all the elements in both arrays.

When the

String()

function is called on the instance, it will return a comma separated string surrounded by brackets. For example,

[1,2,3]

.

Example 1:

Input:

nums = [[1,2],[3,4]], operation = "Add"

Output:

10

Explanation:

const obj1 = new ArrayWrapper([1,2]); const obj2 = new ArrayWrapper([3,4]); obj1 + obj2; // 10

Example 2:

Input:

nums = [[23,98,42,70]], operation = "String"

Output:

"[23,98,42,70]"

Explanation:

const obj = new ArrayWrapper([23,98,42,70]); String(obj); // "[23,98,42,70]"

Example 3:

Input:

nums = [[],[]], operation = "Add"

Output:

0

Explanation:

const obj1 = new ArrayWrapper([]); const obj2 = new ArrayWrapper([]); obj1 + obj2; // 0

Constraints:

0 <= nums.length <= 1000

0 <= nums[i] <= 1000

Note: nums is the array passed to the constructor

Code Snippets

JavaScript:

```
/**
* @param {number[]} nums
* @return {void}
*/
var ArrayWrapper = function(nums) {

};

/**
* @return {number}
*/
ArrayWrapper.prototype.valueOf = function() {

}

/**
* @return {string}
*/
ArrayWrapper.prototype.toString = function() {

}

/**
* const obj1 = new ArrayWrapper([1,2]);
* const obj2 = new ArrayWrapper([3,4]);
* obj1 + obj2; // 10
* String(obj1); // "[1,2]"
* String(obj2); // "[3,4]"
```

```
*/
```

TypeScript:

```
class ArrayWrapper {

constructor(nums: number[]) {

}

valueOf(): number {

}

toString(): string {

}
};

/**
* const obj1 = new ArrayWrapper([1,2]);
* const obj2 = new ArrayWrapper([3,4]);
* obj1 + obj2; // 10
* String(obj1); // "[1,2]"
* String(obj2); // "[3,4]"
*/
```

Solutions

JavaScript Solution:

```
/**
* Problem: Array Wrapper
* Difficulty: Easy
* Tags: array, string
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/
```

```
/**
* @param {number[]} nums
* @return {void}
*/
var ArrayWrapper = function(nums) {

};

/**
* @return {number}
*/
ArrayWrapper.prototype.valueOf = function() {

}

/**
* @return {string}
*/
ArrayWrapper.prototype.toString = function() {

}

/**
* const obj1 = new ArrayWrapper([1,2]);
* const obj2 = new ArrayWrapper([3,4]);
* obj1 + obj2; // 10
* String(obj1); // "[1,2]"
* String(obj2); // "[3,4]"
*/
```

TypeScript Solution:

```
/**
* Problem: Array Wrapper
* Difficulty: Easy
* Tags: array, string
*
* Approach: Use two pointers or sliding window technique
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
```

```
*/

class ArrayWrapper {

constructor(nums: number[]) {

}

valueOf(): number {

}

toString(): string {

}
};

/**
* const obj1 = new ArrayWrapper([1,2]);
* const obj2 = new ArrayWrapper([3,4]);
* obj1 + obj2; // 10
* String(obj1); // "[1,2]"
* String(obj2); // "[3,4]"
*/
```