Problem 2703: Return Length of Arguments Passed

Problem Information

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

Problem Description

Write a function

argumentsLength

that returns the count of arguments passed to it.

Example 1:

Input:

args = [5]

Output:

1

Explanation:

argumentsLength(5); // 1

One value was passed to the function so it should return 1.

Example 2:

Input:

args = [{}, null, "3"]

Output:

3

Explanation:

argumentsLength({}, null, "3"); // 3

Three values were passed to the function so it should return 3.

Constraints:

args

is a valid JSON array

0 <= args.length <= 100

Code Snippets

JavaScript:

```
/**
* @param {...(null|boolean|number|string|Array|Object)} args
* @return {number}
*/
var argumentsLength = function(...args) {

};

/**
* argumentsLength(1, 2, 3); // 3
*/
```

TypeScript:

```
type JSONValue = null | boolean | number | string | JSONValue[] | { [key:
string]: JSONValue };

function argumentsLength(...args: JSONValue[]): number {

};

/**
* argumentsLength(1, 2, 3); // 3
*/
```

Solutions

JavaScript Solution:

```
/**
* Problem: Return Length of Arguments Passed
* Difficulty: Easy
* Tags: array
*
* 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 {...(null|boolean|number|string|Array|Object)} args
* @return {number}
*/
var argumentsLength = function(...args) {

};

/**
* argumentsLength(1, 2, 3); // 3
*/
```

TypeScript Solution:

```
/**
* Problem: Return Length of Arguments Passed
* Difficulty: Easy
* Tags: array
*
* 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
*/

type JSONValue = null | boolean | number | string | JSONValue[] | { [key: string]: JSONValue };

function argumentsLength(...args: JSONValue[]): number {

};

/**
* argumentsLength(1, 2, 3); // 3
*/
```