

Problem 2628: JSON Deep Equal

Problem Information

Difficulty: Medium

Acceptance Rate: 0.00%

Paid Only: No

Problem Description

Given two values

o1

and

o2

, return a boolean value indicating whether two values,

o1

and

o2

, are

deeply equal

For two values to be

deeply equal

, the following conditions must be met:

If both values are primitive types, they are

deeply equal

if they pass the

====

equality check.

If both values are arrays, they are

deeply equal

if they have the same elements in the same order, and each element is also

deeply equal

according to these conditions.

If both values are objects, they are

deeply equal

if they have the same keys, and the associated values for each key are also

deeply equal

according to these conditions.

You may assume both values are the output of

JSON.parse

. In other words, they are valid JSON.

Please solve it without using lodash's

`_ isEqual()`

function

Example 1:

Input:

```
o1 = {"x":1,"y":2}, o2 = {"x":1,"y":2}
```

Output:

true

Explanation:

The keys and values match exactly.

Example 2:

Input:

```
o1 = {"y":2,"x":1}, o2 = {"x":1,"y":2}
```

Output:

true

Explanation:

Although the keys are in a different order, they still match exactly.

Example 3:

Input:

```
o1 = {"x":null,"L":[1,2,3]}, o2 = {"x":null,"L":["1","2","3"]}
```

Output:

false

Explanation:

The array of numbers is different from the array of strings.

Example 4:

Input:

o1 = true, o2 = false

Output:

false

Explanation:

true !== false

Constraints:

1 <= JSON.stringify(o1).length <= 10

5

1 <= JSON.stringify(o2).length <= 10

5

maxNestingDepth <= 1000

Code Snippets

JavaScript:

```

    /**
 * @param {null|boolean|number|string|Array|Object} o1
 * @param {null|boolean|number|string|Array|Object} o2
 * @return {boolean}
 */
var areDeeplyEqual = function(o1, o2) {

};

```

TypeScript:

```

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

function areDeeplyEqual(o1: JSONValue, o2: JSONValue): boolean {
}

```

Solutions

JavaScript Solution:

```

    /**
 * Problem: JSON Deep Equal
 * Difficulty: Medium
 * 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 {null|boolean|number|string|Array|Object} o1
 * @param {null|boolean|number|string|Array|Object} o2
 * @return {boolean}
 */
var areDeeplyEqual = function(o1, o2) {

};

```

TypeScript Solution:

```
/**  
 * Problem: JSON Deep Equal  
 * Difficulty: Medium  
 * 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  
 */  
  
type JSONValue = null | boolean | number | string | JSONValue[] | { [key: string]: JSONValue };  
  
function areDeeplyEqual(o1: JSONValue, o2: JSONValue): boolean {  
};
```