

Problem List

DescriptionEditorialSolutionsSubmissions

2628. JSON Deep EqualPremium

Solved

MediumHint

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 <= 10⁵
- 1 <= JSON.stringify(o2).length <= 10⁵
- maxNestingDepth <= 1000

Seen this question in a real interview before? 1/5

Yes

No

Accepted 12.1K | Submissions 31.4K | Acceptance Rate 38.5%

Hint 1

Hint 2

Hint 3

Similar Questions

Discussion (13)

Copyright © 2024 LeetCode All rights reserved

212131310

</> Code

TypeScriptAuto

```
1 function areDeeplyEqual(o1: any, o2: any): boolean {
2
3   const typeO1 = typeof o1;
4   const typeO2 = typeof o2;
5   if(typeO1 !== typeO2) return false;
6   if(Array.isArray(o1) !== Array.isArray(o2)) return false;
7   if(o1===null || o2===null || typeO1 !== 'object' || typeO2 !== 'object') return o1===o2;
8
9   const o1Keys = Object.keys(o1);
10  const o2Keys = Object.keys(o2);
11  if(o1Keys.length !== o2Keys.length) return false;
12  for(const key of o1Keys) {
13    if(!o2Keys.includes(key)) return false;
14    if(!areDeeplyEqual(o1[key], o2[key])) return false;
15  }
16
17  return true;
18
19 };
```

SavedLn 19, Col 3

TestcaseTest Result

AcceptedRuntime: 71 ms

Case 1

Case 2

Case 3

Case 4

Input