Problem 2705: Compact Object

Problem Information

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

Problem Description

Given an object or array

obj

, return a

compact object

.

A

compact object

is the same as the original object, except with keys containing

falsy

values removed. This operation applies to the object and any nested objects. Arrays are considered objects where the indices are keys. A value is considered

falsy

when

Boolean(value)

returns

false

.

You may assume the

obj

is the output of

JSON.parse

. In other words, it is valid JSON.

Example 1:

Input:

obj = [null, 0, false, 1]

Output:

[1]

Explanation:

All falsy values have been removed from the array.

Example 2:

Input:

obj = {"a": null, "b": [false, 1]}

Output:

{"b": [1]}

Explanation:

obj["a"] and obj["b"][0] had falsy values and were removed.

Example 3:

Input:

obj = [null, 0, 5, [0], [false, 16]]

Output:

[5, [], [16]]

Explanation:

obj[0], obj[1], obj[3][0], and obj[4][0] were falsy and removed.

Constraints:

obj

is a valid JSON object

2 <= JSON.stringify(obj).length <= 10

6

Code Snippets

JavaScript:

```
/**
* @param {Object|Array} obj
* @return {Object|Array}
*/
```

```
var compactObject = function(obj) {

};
```

TypeScript:

```
type JSONValue = null | boolean | number | string | JSONValue[] | { [key:
string]: JSONValue };
type Obj = Record<string, JSONValue> | Array<JSONValue>;

function compactObject(obj: Obj): Obj {

};
```

Solutions

JavaScript Solution:

```
/**
* Problem: Compact Object
* 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 {Object|Array} obj
* @return {Object|Array}
*/
var compactObject = function(obj) {

};
```

TypeScript Solution:

```
/**
* Problem: Compact Object
* 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 };
type Obj = Record<string, JSONValue> | Array<JSONValue>;

function compactObject(obj: Obj): Obj {

};
```