

# Problem 2775: Undefined to Null

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 0.00%

**Paid Only:** No

## Problem Description

Given a deeply nested object or array

obj

, return the object

obj

with any

undefined

values replaced by

null

.

undefined

values are handled differently than

null

values when objects are converted to a JSON string using

JSON.stringify()

. This function helps ensure serialized data is free of unexpected errors.

Example 1:

Input:

```
obj = {"a": undefined, "b": 3}
```

Output:

```
{"a": null, "b": 3}
```

Explanation:

The value for obj.a has been changed from undefined to null

Example 2:

Input:

```
obj = {"a": undefined, "b": ["a", undefined]}
```

Output:

```
{"a": null, "b": ["a", null]}
```

Explanation:

The values for obj.a and obj.b[1] have been changed from undefined to null

Constraints:

obj

is a valid JSON object or array

$2 \leq \text{JSON.stringify(obj).length} \leq 10$

## Code Snippets

### JavaScript:

```
/**
 * @param {Object|Array} obj
 * @return {Object|Array}
 */
var undefinedToNull = function(obj) {

};

/**
 * undefinedToNull({"a": undefined, "b": 3}) // {"a": null, "b": 3}
 * undefinedToNull([undefined, undefined]) // [null, null]
 */
```

### TypeScript:

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

type Obj1 = Record<string, Value> | Array<Value>
type Obj2 = Record<string, JSONValue> | Array<JSONValue>

function undefinedToNull(obj: Obj1): Obj2 {

};

/**
 * undefinedToNull({"a": undefined, "b": 3}) // {"a": null, "b": 3}
 * undefinedToNull([undefined, undefined]) // [null, null]
 */
```

## Solutions

## JavaScript Solution:

```
/**
 * Problem: Undefined to Null
 * 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 undefinedToNull = function(obj) {

};

/**
 * undefinedToNull({"a": undefined, "b": 3}) // {"a": null, "b": 3}
 * undefinedToNull([undefined, undefined]) // [null, null]
 */
```

## TypeScript Solution:

```
/**
 * Problem: Undefined to Null
 * 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 Value = undefined | null | boolean | number | string | Value[] | { [key: string]: Value };

type Obj1 = Record<string, Value> | Array<Value>
```

```
type Obj2 = Record<string, JSONValue> | Array<JSONValue>

function undefinedToNull(obj: Obj1): Obj2 {

};

/**
 * undefinedToNull({"a": undefined, "b": 3}) // {"a": null, "b": 3}
 * undefinedToNull([undefined, undefined]) // [null, null]
 */
```