Problem 2823: Deep Object Filter

Problem Information

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

Problem Description

Given an object or an array

obj

and a function

fn

, return a filtered object or array

filteredObject

.

Function

deepFilter

should perform a deep filter operation on the

obj

. The deep filter operation should remove properties for which the output of the filter function

fn

is

false

, as well as any empty objects or arrays that remain after the keys have been removed.

If the deep filter operation results in an empty object or array, with no remaining properties,

deepFilter

should return

undefined

to indicate that there is no valid data left in the

filteredObject

.

Example 1:

Input:

obj = [-5, -4, -3, -2, -1, 0, 1], fn = (x) => x > 0

Output:

[1]

Explanation:

All values that were not greater than 0 were removed.

Example 2:

Input:

obj = {"a": 1, "b": "2", "c": 3, "d": "4", "e": 5, "f": 6, "g": {"a": 1}}, fn = (x) => typeof x === "string"

Output:

{"b":"2","d":"4"}

Explanation:

All keys with values that were not a string were removed. When the object keys were removed during the filtering process, any resulting empty objects were also removed.

Example 3:

Input:

obj = [-1, [-1, -1, 5, -1, 10], -1, [-1], [-5]], fn = (x) => x > 0

Output:

[[5,10]]

Explanation:

All values that were not greater than 0 were removed. When the values were removed during the filtering process, any resulting empty arrays were also removed.

Example 4:

Input:

obj = [[[[5]]]], fn = (x) => Array.isArray(x)

Output:

undefined

Constraints:

fn

is a function that returns a boolean value

obj

is a valid JSON object or array

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

5

Code Snippets

JavaScript:

```
/**
* @param {Object|Array} obj
* @param {Function} fn
* @return {Object|Array|undefined}
*/
var deepFilter = function(obj, fn) {

};
```

TypeScript:

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

function deepFilter(obj: Obj, fn: Function): Obj | undefined {

};
```

Solutions

JavaScript Solution:

```
/**
* Problem: Deep Object Filter
* 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
* @param {Function} fn
* @return {Object|Array|undefined}
*/
var deepFilter = function(obj, fn) {

};
```

TypeScript Solution:

```
/**
* Problem: Deep Object Filter
* 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 deepFilter(obj: Obj, fn: Function): Obj | undefined {

};
```