Problem 2675: Array of Objects to Matrix

Problem Information

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

Problem Description

Write a function that converts an array of objects

arr

into a matrix

m

.

arr

is an array of objects or arrays. Each item in the array can be deeply nested with child arrays and child objects. It can also contain numbers, strings, booleans, and null values.

The first row

m

should be the column names. If there is no nesting, the column names are the unique keys within the objects. If there is nesting, the column names are the respective paths in the object separated by

"."

.

Each of the remaining rows corresponds to an object in

arr

. Each value in the matrix corresponds to a value in an object. If a given object doesn't contain a value for a given column, the cell should contain an empty string

""

.

The columns in the matrix should be in

lexographically ascending

order.

Example 1:

Input:

arr = [{"b": 1, "a": 2},   {"b": 3, "a": 4}]

Output:

[["a", "b"],   [2, 1],   [4, 3]]

Explanation:

There are two unique column names in the two objects: "a" and "b". "a" corresponds with [2, 4]. "b" coresponds with [1, 3].

Example 2:

Input:

arr = [{"a": 1, "b": 2},   {"c": 3, "d": 4},   {}]

Output:

[["a", "b", "c", "d"],   [1, 2, "", ""],   ["", "", 3, 4],   ["", "", "", ""]]

Explanation:

There are 4 unique column names: "a", "b", "c", "d". The first object has values associated with "a" and "b". The second object has values associated with "c" and "d". The third object has no keys, so it is just a row of empty strings.

Example 3:

Input:

arr = [{"a": {"b": 1, "c": 2}},   {"a": {"b": 3, "d": 4}}]

Output:

[["a.b", "a.c", "a.d"],   [1, 2, ""],   [3, "", 4]]

Explanation:

In this example, the objects are nested. The keys represent the full path to each value separated by periods. There are three paths: "a.b", "a.c", "a.d".

Example 4:

Input:

arr = [[{"a": null}],   [{"b": true}],   [{"c": "x"}]]

Output:

[["0.a", "0.b", "0.c"],   [null, "", ""],   ["", true, ""],   ["", "", "x"]]

Explanation:

Arrays are also considered objects with their keys being their indices. Each array has one element so the keys are "0.a", "0.b", and "0.c".

Example 5:

Input:

arr = [{}, {}, {},]

Output:

[[], [], [], []]

Explanation:

There are no keys so every row is an empty array.

Constraints:

arr

is a valid JSON array

1 <= arr.length <= 1000

unique keys <= 1000

Code Snippets

JavaScript:

```
/**
* @param {Array} arr
* @return {(string | number | boolean | null)[][]}
*/
var jsonToMatrix = function(arr) {

};
```

TypeScript:

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

```
function jsonToMatrix(arr: JSONValue[]): (null | boolean | number |
string)[][] {

};
```

Solutions

JavaScript Solution:

```
/**
* Problem: Array of Objects to Matrix
* Difficulty: Hard
* Tags: array, string, graph
*
* 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 {Array} arr
* @return {(string | number | boolean | null)[][]}
*/
var jsonToMatrix = function(arr) {

};
```

TypeScript Solution:

```
/**
* Problem: Array of Objects to Matrix
* Difficulty: Hard
* Tags: array, string, graph
*
* 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 jsonToMatrix(arr: JSONValue[]): (null | boolean | number |
string)[][] {

};
```