Problem 2722: Join Two Arrays by ID

Problem Information

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

Problem Description

Given two arrays

arr1

and

arr2

, return a new array

joinedArray

. All the objects in each of the two inputs arrays will contain an

id

field that has an integer value.

joinedArray

is an array formed by merging

arr1

and

arr2

based on their

id

key. The length of

joinedArray

should be the length of unique values of

id

. The returned array should be sorted in

ascending

order based on the

id

key.

If a given

id

exists in one array but not the other, the single object with that

id

should be included in the result array without modification.

If two objects share an

id

, their properties should be merged into a single object:

If a key only exists in one object, that single key-value pair should be included in the object.

If a key is included in both objects, the value in the object from

arr2

should override the value from

arr1

.

Example 1:

Input:

arr1 = [{"id": 1, "x": 1},   {"id": 2, "x": 9}], arr2 = [{"id": 3, "x": 5}]

Output:

[{"id": 1, "x": 1},   {"id": 2, "x": 9}, {"id": 3, "x": 5}]

Explanation:

There are no duplicate ids so arr1 is simply concatenated with arr2.

Example 2:

Input:

arr1 = [{"id": 1, "x": 2, "y": 3}, {"id": 2, "x": 3, "y": 6}], arr2 = [{"id": 2, "x": 10, "y": 20}, {"id": 3, "x": 0, "y": 0}]

Output:

[{"id": 1, "x": 2, "y": 3}, {"id": 2, "x": 10, "y": 20},   {"id": 3, "x": 0, "y": 0}]

Explanation:

The two objects with id=1 and id=3 are included in the result array without modifiction. The two objects with id=2 are merged together. The keys from arr2 override the values in arr1.

Example 3:

Input:

arr1 = [{"id": 1, "b": {"b": 94},"v": [4, 3], "y": 48}] arr2 = [{"id": 1, "b": {"c": 84}, "v": [1, 3]}]

Output:

[{"id": 1, "b": {"c": 84}, "v": [1, 3], "y": 48}]

Explanation:

The two objects with id=1 are merged together. For the keys "b" and "v" the values from arr2 are used. Since the key "y" only exists in arr1, that value is taken form arr1.

Constraints:

arr1

and

arr2

are valid JSON arrays

Each object in

arr1

and

arr2

has a unique integer

id

key

2 <= JSON.stringify(arr1).length <= 10

6

2 <= JSON.stringify(arr2).length <= 10

6

Code Snippets

JavaScript:

```
/**
* @param {Array} arr1
* @param {Array} arr2
* @return {Array}
*/
var join = function(arr1, arr2) {

};
```

TypeScript:

```
type JSONValue = null | boolean | number | string | JSONValue[] | { [key:
string]: JSONValue };
type ArrayType = { "id": number } & Record<string, JSONValue>;

function join(arr1: ArrayType[], arr2: ArrayType[]): ArrayType[] {

};
```

Solutions

JavaScript Solution:

```
/**
* Problem: Join Two Arrays by ID
* Difficulty: Medium
* Tags: array, string, sort
*
* 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} arr1
* @param {Array} arr2
* @return {Array}
*/
var join = function(arr1, arr2) {

};
```

TypeScript Solution:

```
/**
* Problem: Join Two Arrays by ID
* Difficulty: Medium
* Tags: array, string, sort
*
* 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 ArrayType = { "id": number } & Record<string, JSONValue>;

function join(arr1: ArrayType[], arr2: ArrayType[]): ArrayType[] {

};
```