Problem 2822: Inversion of Object

Problem Information

Difficulty: Easy
Acceptance Rate: 59.96%
Paid Only: Yes

Problem Description

Given an object or an array `obj`, return an inverted object or array `invertedObj`.

The `invertedObj` should have the keys of `obj` as values and the values of `obj` as keys. The indices of array should be treated as keys.

The function should handle duplicates, meaning that if there are multiple keys in `obj` with the same value, the `invertedObj` should map the value to an array containing all corresponding keys.

It is guaranteed that the values in `obj` are only strings.

Example 1:

Input: obj = {"a": "1", "b": "2", "c": "3", "d": "4"} **Output:** invertedObj = {"1": "a", "2": "b", "3": "c", "4": "d"} **Explanation:** The keys from obj become the values in invertedObj, and the values from obj become the keys in invertedObj.

Example 2:

Input: obj = {"a": "1", "b": "2", "c": "2", "d": "4"} **Output:** invertedObj = {"1": "a", "2": ["b", "c"], "4": "d"} **Explanation:** There are two keys in obj with the same value, the invertedObj mapped the value to an array containing all corresponding keys.

Example 3:

Input: obj = ["1", "2", "3", "4"] **Output:** invertedObj = {"1": "0", "2": "1", "3": "2", "4": "3"} **Explanation:** Arrays are also objects therefore array has changed to an object and the keys (indices) from obj become the values in invertedObj, and the values from obj become the keys in invertedObj.

Constraints:

* `obj` is a valid JSON object or array * `typeof obj[key] === "string"` * `2 <= JSON.stringify(obj).length <= 105`

Code Snippets

JavaScript:

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

};
```

TypeScript:

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

function invertObject(obj: Obj): Record<string, JSONValue> {

};
```