Problem 2633: Convert Object to JSON String

Problem Information

Difficulty: Medium
Acceptance Rate: 77.87%
Paid Only: Yes

Problem Description

Given a value, return a valid JSON string of that value. The value can be a string, number, array, object, boolean, or null. The returned string should not include extra spaces. The order of keys should be the same as the order returned by `Object.keys()`.

Please solve it without using the built-in `JSON.stringify` method.

Example 1:

Input: object = {"y":1,"x":2} **Output:** {"y":1,"x":2} **Explanation:** Return the JSON representation. Note that the order of keys should be the same as the order returned by Object.keys().

Example 2:

Input: object = {"a":"str","b":-12,"c":true,"d":null} **Output:** {"a":"str","b":-12,"c":true,"d":null} **Explanation:** The primitives of JSON are strings, numbers, booleans, and null.

Example 3:

Input: object = {"key":{"a":1,"b":[{},null,"Hello"]}} **Output:** {"key":{"a":1,"b":[{},null,"Hello"]}} **Explanation:** Objects and arrays can include other objects and arrays.

Example 4:

Input: object = true **Output:** true **Explanation:** Primitive types are valid inputs.

Constraints:

* `value` is a valid JSON value * `1 <= JSON.stringify(object).length <= 105` * `maxNestingLevel <= 1000` * all strings contain only alphanumeric characters

Code Snippets

JavaScript:

```
/**
* @param {null|boolean|number|string|Array|Object} object
* @return {string}
*/
var jsonStringify = function(object) {

};
```

TypeScript:

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

function jsonStringify(object: JSONValue): string {

};
```