JavaScript Same Tree

Challenge

Given the roots of two binary trees `p` and `q`, write a function to check if they are the same or not.

Two binary trees are considered the same if they are structurally identical, and the nodes have the same value.

1st Example

```
Input: p = [1,2,3], q = [1,2,3]
Output: true
```

2nd Example

```
Input: p = [1,2], q = [1,null,2]
Output: false
```

3rd Example

```
Input: p = [1,2,1], q = [1,1,2]
Output: false
```

Constraints

- `-10⁴ <= Node.val <= 10⁴`
- The number of nodes in both trees is in the range `[0, 100]`.

Solution

```
const isSameTree = (p, q) => {
    return JSON.stringify(p) === JSON.stringify(q);
};
```

Explanation

I've coded a function called `isSameTree` that takes two arguments, `p` and `q`. The purpose of this function is to compare if the JSON stringified versions of the two arguments are equal.

Inside the function, the `JSON.stringify()` method is used to convert both `p` and `q` into JSON strings. This method serializes the arguments into a string representation.

The `===` operator is then used to compare the two JSON strings. This operator checks for strict equality, meaning it not only compares the values but also the types of the values.

If the two JSON strings are equal, which means the arguments have the same structure and values, the function returns `true`.

Otherwise, it returns `false`, indicating that the arguments are different.

In summary, this function compares if two arguments, `p` and `q`, are the same by converting them into JSON strings and checking for equality. It provides a convenient way to compare complex data structures represented as objects or arrays.

Author: Trevor Morin