JavaScript Set Mismatch

Challenge

You have a set of integers `s`, which originally contains all the numbers from `1` to `n`. Unfortunately, due to some error, one of the numbers in `s` got duplicated to another number in the set, which results in repetition of one number and loss of another number.

You are given an integer array `nums` representing the data status of this set after the error.

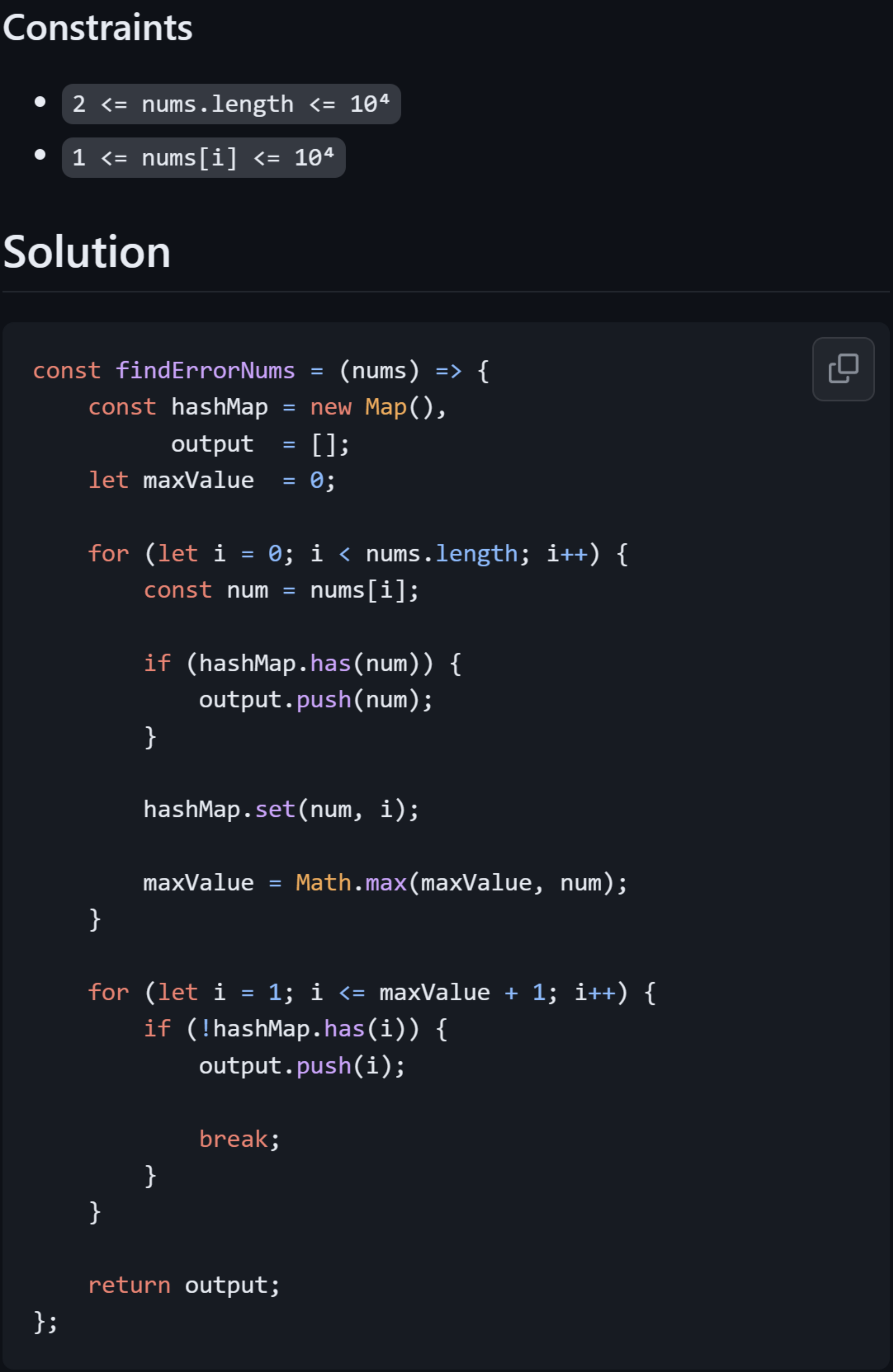
Find the number that occurs twice and the number that is missing and return them in the form of an array.

1st Example

```
Input: nums = [1,2,2,4]
Output: [2,3]
```

2nd Example

```
Input: nums = [1,1]
Output: [1,2]
```



Constraints

- 2 <= nums.length <= 10^4
- 1 <= nums[i] <= 10^4

Solution

```
const findErrorNums = (nums) => {
    const hashMap = new Map(),
          output   = [];
    let maxValue  = 0;

    for (let i = 0; i < nums.length; i++) {
        const num = nums[i];

        if (hashMap.has(num)) {
            output.push(num);
        }

        hashMap.set(num, i);

        maxValue = Math.max(maxValue, num);
    }

    for (let i = 1; i <= maxValue + 1; i++) {
        if (!hashMap.has(i)) {
            output.push(i);

            break;
        }
    }

    return output;
};
```

Explanation

I've defined a function called `findErrorNums` that takes an array of numbers as its input and returns an array containing two elements.

The function first initializes an empty hash map and an empty array called `output`. It also sets a variable `maxValue` to `0`.

Then, it loops through each element in the `nums` array. For each element, it checks if it already exists in the hash map. If it does, it means that the element is a duplicate, so it is pushed into the `output` array. Then, the element is added to the hash map with its index as the value. The `maxValue` variable is updated to keep track of the largest number in the array.

After the first loop, it enters a second loop that starts from `1` and goes up to `maxValue + 1`. It checks if each number from `1` to `maxValue + 1` exists in the hash map. If it doesn't, it means that the number is missing from the array, so it is pushed into the `output` array. The loop then breaks.

Finally, the `output` array is returned.

In summary, this function finds and returns an array containing one duplicate number and one missing number from the input array.

Author: Trevor Morin