JavaScript Intersection of Two Arrays

Challenge

Given two integer arrays `nums1` and `nums2`, return an array of their intersection. Each element in the result must be unique and you may return the result in any order.

1st Example

```
Input: nums1 = [1,2,2,1], nums2 = [2,2]
Output: [2]
```

2nd Example

```
Input: nums1 = [4,9,5], nums2 = [9,4,9,8,4]
Output: [9,4]
Explanation: [4,9] is also accepted.
```

Constraints

- `1 <= nums1.length, nums2.length <= 1000`
- `0 <= nums1[i], nums2[i] <= 1000`

Solution

```
const intersection = (nums1, nums2) => {
    const output  = [],
          hashmap = new Map();

    for (let i = 0; i < nums1.length; i++) {
        hashmap.set(nums1[i], i);
    }

    for (let i = 0; i < nums2.length; i++) {
        const num = nums2[i];

        if (hashmap.has(num)) {
            output.push(num);

            hashmap.delete(num);
        }
    }

    return output;
};
```

Explanation

I've built a function called `intersection` that takes in two arrays, `nums1` and `nums2`, as parameters. Its purpose is to find the common elements between the two arrays and return them in a new array called `output`.

Inside the function, an empty array called `output` is created to

store the common elements. Additionally, a new `Map` object called `hashmap` is initialized.

The function then iterates through each element in `nums1` using a for loop. For each element, it uses the `set()` method of the `Map` object to store the element as the key and its index as the value.

Next, the function iterates through each element in `nums2` using another for loop. For each element, it assigns it to a variable called `num`. It then checks if the `hashmap` contains the `num` as a key using the `has()` method.

If the `hashmap` contains the `num`, it means that it is a common element between `nums1` and `nums2`. The function then pushes the `num` into the `output` array using the `push()` method and removes the `num` from the `hashmap` using the `delete()` method.

Finally, after iterating through all elements in `nums2`, the function returns the `output` array, which contains the common elements between `nums1` and `nums2`.

In summary, the `intersection` function finds the common elements between two arrays (`nums1` and `nums2`) by utilizing a `Map` object. It stores the elements of the first array in the map and then checks for common elements in the second array, pushing them into an output array.

Author: Trevor Morin