1001 N-Repeated Element in Size 2N Array (link)

Description

You are given an integer array `nums` with the following properties:

- `nums.length == 2 * n`.
- `nums` contains `n + 1` **unique** elements.
- Exactly one element of `nums` is repeated `n` times.

Return *the element that is repeated* `n` *times*.

Example 1:

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

Example 2:

```
Input: nums = [2,1,2,5,3,2]
Output: 2
```

Example 3:

```
Input: nums = [5,1,5,2,5,3,5,4]
Output: 5
```

Constraints:

- `2 <= n <= 5000`
- `nums.length == 2 * n`
- `0 <= nums[i] <= 10^4`
- `nums` contains `n + 1` **unique** elements and one of them is repeated exactly `n` times.

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
#include <unordered_map>
using namespace std;

class Solution {
public:
    int repeatedNTimes(vector<int>& nums) {
        unordered_map<int, int> count;

        for (int num : nums) {
            count[num]++;
            if (count[num] > 1) {
                return num;
            }
        }

        return -1;
    }
};
```