950 X of a Kind in a Deck of Cards (link)

Description

You are given an integer array `deck` where `deck[i]` represents the number written on the i^{th} card.

Partition the cards into **one or more groups** such that:

- Each group has **exactly** `x` cards where `x > 1`, and
- All the cards in one group have the same integer written on them.

Return `true` *if such partition is possible, or* `false` *otherwise*.

Example 1:

```
Input: deck = [1,2,3,4,4,3,2,1]
Output: true
Explanation: Possible partition [1,1],[2,2],[3,3],[4,4].
```

Example 2:

```
Input: deck = [1,1,1,2,2,2,3,3]
Output: false
Explanation: No possible partition.
```

Constraints:

- `1 <= deck.length <= 10^4`
- `0 <= deck[i] < 10^4`

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
#include <unordered_map>
#include <algorithm>

class Solution {
public:
    bool hasGroupsSizeX(std::vector<int>& deck) {
        std::unordered_map<int, int> count;

        // Считаем частоту каждой карты в массиве
        for (int card : deck) {
            count[card]++;
        }

        // Находим НОД всех частот
        int gcd_value = 0;
        for (auto& pair : count) {
            gcd_value = gcd(gcd_value, pair.second);
        }

        // Если НОД больше 1, то можно разделить карты на группы
        return gcd_value > 1;
    }

private:
    // Функция для вычисления НОД двух чисел
    int gcd(int a, int b) {
        return b == 0 ? a : gcd(b, a % b);
    }
};
```