

914. X of a Kind in a Deck of Cards

Solved

Easy

Topics

Companies

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.

The screenshot displays a code editor interface for the problem '914. X of a Kind in a Deck of Cards'. The left sidebar shows the problem description and a submission status of 'Accepted' for a user named 'Zhukovilya' submitted at Jun 30, 2024 10:39. The submission details include a runtime of 100 ms (Beats 72.29%) and memory usage of 47.56 MB (Beats 61.45%). A bar chart shows the distribution of runtime results. The main editor area displays the C# code for the solution, which uses a Dictionary to count the frequency of each card value and then checks if the frequency is divisible by the group size 'x'.

```

1 public class Solution {
2     public bool HasGroupsSizeX(int[] deck) {
3         // Считаем частоту каждого числа в массиве
4         Dictionary<int, int> count = new Dictionary<int, int>();
5         foreach (int card in deck) {
6             if (count.ContainsKey(card)) {
7                 count[card]++;
8             } else {
9                 count[card] = 1;
10            }
11        }
12    }
13 }

```

Код:

```

public class Solution
{
    public bool HasGroupsSizeX(int[] deck)
    {
        // Считаем частоту каждого числа в массиве
        Dictionary<int, int> count = new Dictionary<int, int>();
        foreach (int card in deck)
    
```

```

    {
        if (count.ContainsKey(card))
        {
            count[card]++;
        }
        else
        {
            count[card] = 1;
        }
    }

    // Находим наименьший общий делитель для всех частот
    int gcd = count.Values.First();
    foreach (int val in count.Values)
    {
        gcd = GCD(gcd, val);
        if (gcd == 1)
        {
            return false; // если НОД равен 1, то невозможно разделить карты на
группы > 1
        }
    }

    return gcd > 1;
}

// Метод для нахождения НОД (Алгоритм Евклида)
private int GCD(int a, int b)
{
    while (b != 0)
    {
        int temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}
}

```