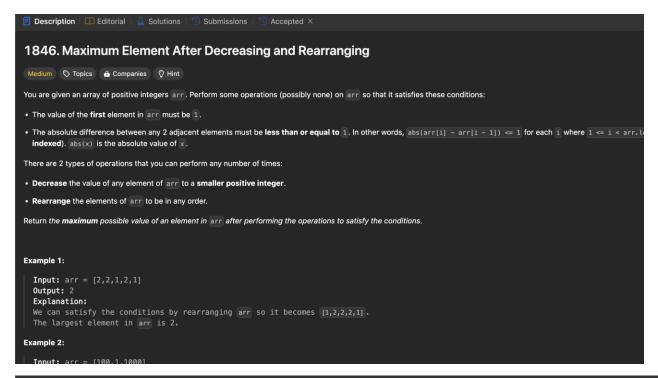# MCS – 253P ADVANCED PROGRAMMING AND PROBLEM SOLVING

LAB 7 Program(Maximum Element After Decreasing and Rearranging)

Aswin Sampath

saswin@uci.edu

53844684

Question:)

Description | Editorial | Solutions | Submissions | Accepted ×

1846. Maximum Element After Decreasing and Rearranging

Medium | Topics | Companies | Hint

You are given an array of positive integers `arr`. Perform some operations (possibly none) on `arr` so that it satisfies these conditions:

- The value of the **first** element in `arr` must be `1`.

- The absolute difference between any 2 adjacent elements must be **less than or equal to** `1`. In other words, `abs(arr[i] - arr[i - 1]) <= 1` for each `i` where `1 <= i < arr.le` **indexed**). `abs(x)` is the absolute value of `x`.

There are 2 types of operations that you can perform any number of times:

- **Decrease** the value of any element of `arr` to a **smaller positive integer**.

- **Rearrange** the elements of `arr` to be in any order.

Return the **maximum** possible value of an element in `arr` after performing the operations to satisfy the conditions.

Example 1:

```
Input: arr = [2,2,1,2,1]
Output: 2
Explanation:
We can satisfy the conditions by rearranging arr so it becomes [1,2,2,2,1].
The largest element in arr is 2.
```

Example 2:

```
Input: arr = [100,1,1000]
```

Description | Editorial | Solutions | Submissions | Accepted ×

```
We can satisfy the conditions by rearranging arr so it becomes [1,2,2,2,1].
The largest element in arr is 2.
```

Example 2:

```
Input: arr = [100,1,1000]
Output: 3
Explanation:
One possible way to satisfy the conditions is by doing the following:
1. Rearrange arr so it becomes [1,100,1000].
2. Decrease the value of the second element to 2.
3. Decrease the value of the third element to 3.
Now arr = [1,2,3], which satisfies the conditions.
The largest element in arr is 3.
```

Example 3:

```
Input: arr = [1,2,3,4,5]
Output: 5
Explanation: The array already satisfies the conditions, and the largest element is 5.
```

Constraints:

- $1 <= arr.length <= 10^5$

- $1 <= arr[i] <= 10^9$

Code:

```cpp
class Solution {
public:
    int maximumElementAfterDecrementingAndRearranging(vector<int>& arr) {
        int desiredNumber=1;
        sort(arr.begin(),arr.end());
        arr[0]=1;
        for(int i=1;i<arr.size();i++){
            if(arr[i]-arr[i-1]<=1)continue;
            arr[i]=arr[i-1]+1;
        }
        return arr[arr.size()-1];
    }
};
```

Output: