

6955. Maximum Number of Groups With Increasing Length

My Submissions (/contest/weekly-contest-355/problems/maximum-number-of-groups-with-increasing-length/submissions/)

Back to Contest (/contest/weekly-contest-355/)

You are given a **0-indexed** array `usageLimits` of length `n`.

Your task is to create **groups** using numbers from `0` to `n - 1`, ensuring that each number, `i`, is used no more than `usageLimits[i]` times in total **across all groups**. You must also satisfy the following conditions:

- Each group must consist of **distinct** numbers, meaning that no duplicate numbers are allowed within a single group.
- Each group (except the first one) must have a length **strictly greater** than the previous group.

Return an integer denoting the **maximum** number of groups you can create while satisfying these conditions.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Hard

Example 1:

**Input:** `usageLimits = [1,2,5]`  
**Output:** `3`  
**Explanation:** In this example, we can use `0` at most once, `1` at most twice, and `2` at most five times. One way of creating the maximum number of groups while satisfying the conditions is:  
Group 1 contains the number `[2]`.  
Group 2 contains the numbers `[1,2]`.  
Group 3 contains the numbers `[0,1,2]`.  
It can be shown that the maximum number of groups is 3.  
So, the output is 3.

Example 2:

**Input:** `usageLimits = [2,1,2]`  
**Output:** `2`  
**Explanation:** In this example, we can use `0` at most twice, `1` at most once, and `2` at most twice. One way of creating the maximum number of groups while satisfying the conditions is:  
Group 1 contains the number `[0]`.  
Group 2 contains the numbers `[1,2]`.  
It can be shown that the maximum number of groups is 2.  
So, the output is 2.

Example 3:

**Input:** `usageLimits = [1,1]`  
**Output:** `1`  
**Explanation:** In this example, we can use both `0` and `1` at most once. One way of creating the maximum number of groups while satisfying the conditions is:  
Group 1 contains the number `[0]`.  
It can be shown that the maximum number of groups is 1.  
So, the output is 1.

Constraints:

- `1 <= usageLimits.length <= 105`
- `1 <= usageLimits[i] <= 109`

JavaScript

```
1 const maxIncreasingGroups = (a) => {
2   a.sort((x, y) => x - y);
3   let cur = 0, res = 0;
4   for (const x of a) {
5     cur += x;
```

```
6 ▼      if (cur >= res + 1) {  
7          res++;  
8          cur -= res;  
9      }  
10     }  
11     return res;  
12 };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)**Submission Result: Accepted** (</submissions/detail/1001544020/>) [More Details >](/submissions/detail/1001544020/)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support/\)](/support/) | [Jobs \(/jobs/\)](/jobs/) | [Bug Bounty \(/bugbounty/\)](/bugbounty/) | [Online Interview \(/interview/\)](/interview/) | [Students \(/student/\)](/student/) | [Terms \(/terms/\)](/terms/) | [Privacy Policy \(/privacy/\)](/privacy/) [United States \(/region/\)](/region/)