Problem 2358: Maximum Number of Groups Entering a Competition

Problem Information

Difficulty: Medium
Acceptance Rate: 68.40%
Paid Only: No
Tags: Array, Math, Binary Search, Greedy

Problem Description

You are given a positive integer array `grades` which represents the grades of students in a university. You would like to enter **all** these students into a competition in **ordered** non-empty groups, such that the ordering meets the following conditions:

* The sum of the grades of students in the `ith` group is **less than** the sum of the grades of students in the `(i + 1)th` group, for all groups (except the last). * The total number of students in the `ith` group is **less than** the total number of students in the `(i + 1)th` group, for all groups (except the last).

Return _the**maximum** number of groups that can be formed_.

Example 1:

Input: grades = [10,6,12,7,3,5] **Output:** 3 **Explanation:** The following is a possible way to form 3 groups of students: - 1st group has the students with grades = [12]. Sum of grades: 12. Student count: 1 - 2nd group has the students with grades = [6,7]. Sum of grades: 6 + 7 = 13. Student count: 2 - 3rd group has the students with grades = [10,3,5]. Sum of grades: 10 + 3 + 5 = 18. Student count: 3 It can be shown that it is not possible to form more than 3 groups.

Example 2:

Input: grades = [8,8] **Output:** 1 **Explanation:** We can only form 1 group, since forming 2 groups would lead to an equal number of students in both groups.

Constraints:

* `1 <= grades.length <= 105` * `1 <= grades[i] <= 105`

Code Snippets

C++:

```
class Solution {
public:
int maximumGroups(vector<int>& grades) {

}
};
```

Java:

```
class Solution {
public int maximumGroups(int[] grades) {

}
}
```

Python3:

```
class Solution:
def maximumGroups(self, grades: List[int]) -> int:
```