

Allocate Minimum Pages

Difficulty: Medium

Accuracy: 35.51%

Submissions: 243K+

Points: 4

You are given an array `arr[]` of integers, where each element `arr[i]` represents the number of pages in the `i`th book. You also have an integer `k` representing the number of students. The task is to allocate books to each student such that:

- Each student receives **atleast** one book.
- Each student is assigned a contiguous sequence of books.
- No book is assigned to more than one student.

The objective is to minimize the maximum number of pages assigned to any student. In other words, out of all possible allocations, find the arrangement where the student who receives the most pages still has the smallest possible maximum.

Note: Return **-1** if a valid assignment is not possible, and allotment should be in contiguous order (see the explanation for better understanding).

Examples:

Input: `arr[] = [12, 34, 67, 90]`, `k = 2`

Output: 113

Explanation: Allocation can be done in following ways:

```
1 // } Driver Code Ends
30 class Solution {
31     public static boolean check(int[] arr, int k , int pageLimit){
32         int cnt = 1;
33         int pageSum = 0;
34         for(int i = 0; i < arr.length; i++){
35             if(pageSum + arr[i] > pageLimit){
36                 cnt++;
37                 pageSum = arr[i] ;
38             }
39             else {
40                 pageSum += arr[i];
41             }
42         }
43         return (cnt <= k);
44     }
45
46     public static int findPages(int[] arr, int k) {
47         if(k > arr.length)
48             return -1;
49         int lo = Arrays.stream(arr).max().getAsInt();
50         int hi = Arrays.stream(arr).sum();
51         int res = -1;
52         while (lo <= hi){
53             int mid = lo + (hi - lo) / 2;
54             if(check(arr,k,mid)){
55                 res = mid;
56                 hi = mid - 1;
57             }
58             else {
59                 lo = mid + 1;
60             }
61         }
62         return res;
63     }
64 }
```