BOOK ALLOCATION

☆ In this problem, we are given an array of integers where each integer represents number of pages in the i^{th} book along with m students

Our job is to allocate books such that :

- Each student gets atleast 1 book
- Each book is alotted to at max 1 student
- Books are assigned in continous order.

Our job is to find out if the maximum number of pages alotted to a student in a particular configuration is minimum among all configurations or not.

Brute force solution is to iterate through all numbers from max to sum, and then check if each student is able to get a book or not. Optimal solution is to do this using Binary Search.

Pseudocode :

```
pagesChecker(arr, N, m, s) {
    k = 1, cu = 0
```

```
    for (i = 1 → N) {
        if (arr[i] + cu > m) {
            k++
            cu = arr[i]
        } else cu += arr[i]
    }
    return k
}
bookAllocation(arr, N, S) {
    max, sum = 0, 0
    for (i = 0 → N) {
        if (arr[i] > max) {
            max = arr[i]
        }
        sum += arr[i]
    }
    ans = -1
    while (max <= sum) {
        mid = (max + sum)/2
        ret = pagesChecker(arr, N, mid, S)
        if (ret < S) {
            sum = mid - 1
        } else {
            max = mid + 1
        }
    }
    return max
}
```