Allocate minimum number of pages

You are given **N** number of books. Every i^{th} book has $\mathbf{A_i}$ number of pages.

You have to allocate contagious books to **M** number of students. There can be many ways or permutations to do so. In each permutation, one of the **M** students will be allocated the maximum number of pages. Out of all these permutations, the task is to find that particular permutation in which the maximum number of pages allocated to a student is minimum of those in all the other permutations and print this minimum value.

Each book will be allocated to exactly one student. Each student has to be allocated at least one book.

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

Example 1:

```
Input: N = 4
A[] = {12,34,67,90}
M = 2
Output: 113 Explanation:
Allocation can be done in following ways:
{12} and {34, 67, 90} Maximum Pages = 191
{12, 34} and {67, 90} Maximum Pages = 157
{12, 34, 67} and {90}  Maximum Pages =113
Therefore, the minimum of these cases is
113, which is selected as the output.
```

Example 2:

```
Input: N = 3
A[] = {15,17,20}
M = 2
Output: 32 Explanation: Allocation is done as
{15,17} and {20}
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **findPages()** which takes 2 Integers N, and m and an array A[] of length N as input and returns the expected answer.

Expected Time Complexity: O(NlogN)
Expected Auxilliary Space: O(1)

Constraints:

$1 <= N <= 10^5$

$1 <= A[i] <= 10^6$

$1 <= M <= 10^5$

```
def findPages(self,A, N, M):
        #code here
        if N==M:
            return max(A)
        if N<M:
            return -1
        lo = max(A)
        hi = sum(A)
        ans = -1
        while lo<=hi:
            mid = (lo+hi)//2

            if self.isValid(A,mid,M):
                ans = mid
                hi = mid-1
            else:
                lo = mid+1
        return ans


    def isValid(self,A,pages,M):
        tempSum = 0
        students = 1

        for i in range(len(A)):
            tempSum = tempSum+A[i]
            if tempSum>pages:
                students+=1
                tempSum = A[i]

        if students>M:
            return False
        else:
            return True
```