**Approach : Variant of binary search**

[(https://www.youtube.com/watch?v=2JSQIhPcHQg)](https://www.youtube.com/watch?v=2JSQIhPcHQg))

Eg : 10 20 30 40 N=4 M=2(no of students)

1. Let start be initialized to max in array(max no of pages an individual can read provided he/she is given 1 book).

START = 40

1. Let end be initialized to sum of array.

END = 100

1. While start <= end
   1. Find mid = start + end /2
   2. Mid denotes max books we should allow any student to ready
   3. Now check if Mid is a valid split point that we chose, to understand if really by allowing only mid no of books to each student can we make a successful arrangement
   4. If yes, add to result. And move left to find lesser value(to minimize)

END = mid-1

* 1. Otherwise, move to right(maybe current capacity no enough to distribute amongst all students, increase capacity)

1. Finally return result

isValid(A, N, M,Midmax)

1. Assume no of student = 1
2. Iterate over books array:
   1. Keep adding no of pages to sum
   2. If sum > Midmax/capacity:
      1. Increase no of students
      2. Reinitialize sum to current value
   3. If no of students exceeds given M:
      1. Return False
3. Return True

CODE: