[12, 34, 67, 90]   M = 2

[12] [34, 67, 90] → 12, 191

[12, 34] [67, 90] → 46, 157

✓ [12, 34, 67] [90] → 113, 90

[a0, a1, a2, a3, a4 ...]   M

M-1

[90, 203]
90 ↓ M=4, 203 ↓ M=1

M = N
res = max element in the array

M = 1
res = array sum.

[10, 20, 10, 30]   M = 2

Search space : [30, 70] (res ↑)
M = 4, 3, 2, 1 M = 1 (M↓)

Find the no. of students s.t. no student gets more than 'mid' pages

low = 30   high = 70
mid = 50

[10, 20, 10, 30] (1, 1, 1, 2)

res = ~~50~~ ~~44~~ ~~41~~ 40

Discard the right half

low = 30   high = 49
mid = 79/2 = 39

[10, 20, 10, 30] (1, 1, 2, 3)

Discard the left half

low = 40   high = 49
mid = 89/2 = 44

[10, 20, 10, 30] (1, 1, 1, 2)

Discard the right half

low = 40   high = 43
mid = 83/2 = 41

[10, 20, 10, 30] (1, 1, 1, 2)

Discard the right half

low = 40   high = 40
mid = 40.

[10, 20, 10, 30] (1, 1, 1, 2)

Discard the right half

low = 40   high = 39

TC: O(n * log(sum))
AS: O(1).

```cpp
bool isPossible(vector<int> &pages, int M, int mid) {

    // [10, 20, 10, 30] mid = 39
    int studentCount = 0, pagesAllocated = 0;
    for (int i = 0; i < pages.size(); i++) {
        if (pagesAllocated + pages[i] <= mid) {
            pagesAllocated += pages[i];
        } else {
            studentCount++;
            pagesAllocated = pages[i];
        }
    }

    return studentCount <= M;
}
```

M = 2   mid = 39
[10, 20, 10, 30]

~~SC = 0~~ 1   PA = ~~0~~ ~~10~~ 30 ~~70~~

SC = ~~1~~ ~~2~~ 3   PA = ~~0~~ ~~10~~ ~~20~~ ~~10~~ 30.