

DSAweek4HW - Google Drive x Allocate minimum number of p x +

practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

Gmail YouTube Maps

Problems Courses Get Hired Contests </>POTD


æ

130 🔍 🔔 👤

⌵ </> Problem Editorial ⌚ Submissions 💬 Comments

Allocate minimum number of pages

Hard Accuracy: 35.51% Submissions: 115K+ Points: 8

 Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now! [↗](#)

You are given **N** number of books. Every **i**th book has **A_i** number of pages.
You have to allocate contiguous 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 the 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

C++ (g++ 5.4) Average Time: 35m Start Timer ⌵

```
1 // } Driver Code Ends
2 //User function template in C++
3
4
5
6
7
8
9
10
11 class Solution
12 {
13     public:
14         //Function to find minimum number of pages.
15         int findPages(int A[], int N, int M)
16         {
17
18         }
19 };
20 // } Driver Code Ends
```

⌵



DSAWeek4HW - Google Drive x Allocate minimum number of p x +


← → ↺ practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

Gmail YouTube Maps

Problems Courses Get Hired Contests </>POTD

⌵

⌵ Problem Editorial Submissions Comments

 Don't Miss Out on the Chance to Work with Leading Companies! Visit the GFG Job Fair Now! ↗

You are given **N** number of books. Every **i**th book has **A_i** number of pages.
You have to allocate contiguous 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 the 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,

æ


⚡ 130 🔍 🔔 👤

C++ (g++ 5.4) Average Time: 35m Start Timer ⌵

```
1 // } Driver Code Ends
9 //User function template in C++
10
11 class Solution
12 {
13     public:
14         //Function to find minimum number of pages.
15         int findPages(int A[], int N, int M)
16         {
17
18         }
19     };
20 // } Driver Code Ends
```

⌵

💡



Book Allocation Problem

12	34	67	90
0	1	2	3

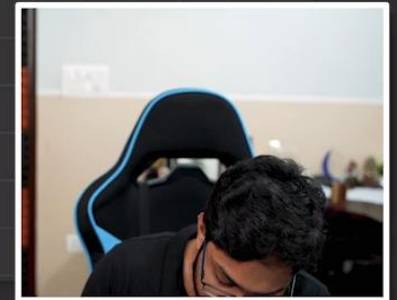
array of book
→ $A_i \rightarrow$ no. of
Page in that
Book_i

$M =$ learn.thecodehelp.in
jhe6587@gmail.com
+919315347281

①
$$\begin{array}{c} 12 \\ \underline{0} \end{array} \mid \begin{array}{c} 34 \quad 67 \quad 90 \\ \underline{1} \end{array}$$

②
$$\begin{array}{c} 12 \quad 34 \\ \underline{0} \end{array} \mid \begin{array}{c} 67 \quad 90 \\ \underline{1} \end{array}$$

③
$$\begin{array}{c} 12 \quad 34 \quad 67 \\ \underline{0} \end{array} \mid \begin{array}{c} 90 \\ \underline{1} \end{array}$$



Book Allocation Problem

12	34	67	90
0	1	2	3

array of book
→ $A_i \rightarrow$ no. of
Page in that
Book,

$M = 2,$

①

<u>12</u>		<u>191</u>
12	34	67 90
0		1

→ Max. 191

②

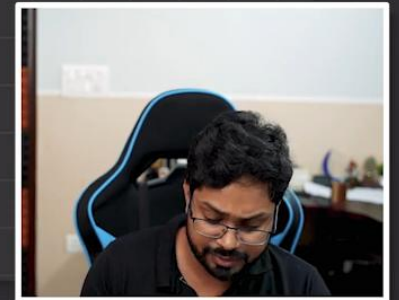
<u>46</u>		<u>157</u>
12 34	67	90
0		1

→ Max. 157

③

<u>113</u>		<u>90</u>
12 34 67	90	
0	1	

→ max. 113,



Book Allocation Problem

12	34	67	90
0	1	2	3

array of book
→ $A_i \rightarrow$ no. of
Page in that
Book,

$M = 2,$

①
$$\begin{array}{r|l} \begin{array}{r} 12 \\ \hline 12 \\ 0 \end{array} & \begin{array}{r} 191 \\ \hline 34 \quad 67 \quad 90 \\ \hline 1 \end{array} \end{array}$$

→ Max. 191

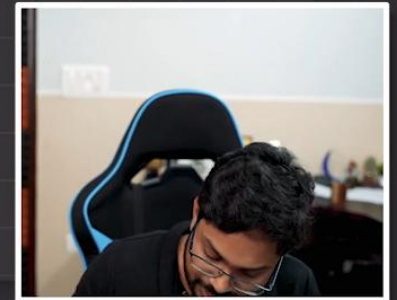
②
$$\begin{array}{r|l} \begin{array}{r} 46 \\ \hline 12 \quad 34 \\ 0 \end{array} & \begin{array}{r} 157 \\ \hline 67 \quad 90 \\ \hline 1 \end{array} \end{array}$$

→ Max. 157

③
$$\begin{array}{r|l} \begin{array}{r} 113 \\ \hline 12 \quad 34 \quad 67 \\ 0 \end{array} & \begin{array}{r} 90 \\ \hline 90 \\ 1 \end{array} \end{array}$$

→ max. 113.

113



DSAweek4HW - Google Drive x Allocate minimum number of p. x +

practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

Gmail YouTube Maps

Problems Courses Get Hired Contests </>POTD

⌵

Problem

Editorial

Submissions

Comments

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

æG

130 🔍 🔔 👤

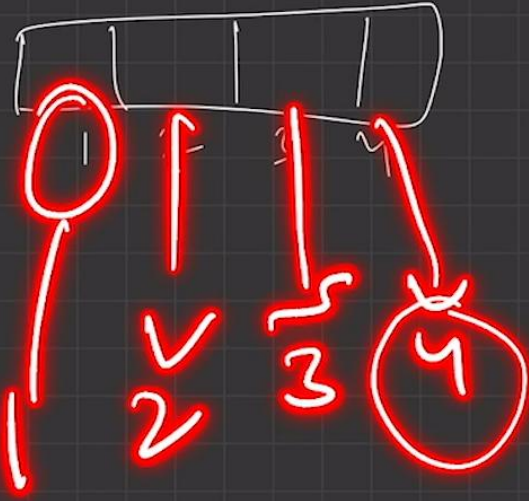
C++ (g++ 5.4) Average Time: 35m Start Timer

```
1 // } Driver Code Ends
9 //User function template in C++
10
11 class Solution
12 {
13     public:
14         //Function to find minimum number of pages.
15         int findPages(int A[], int N, int M)
16         {
17
18         }
19     };
20 // } Driver Code Ends
```

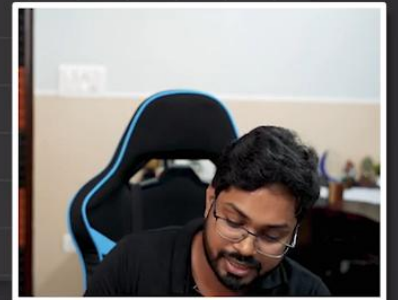
⌵

💡

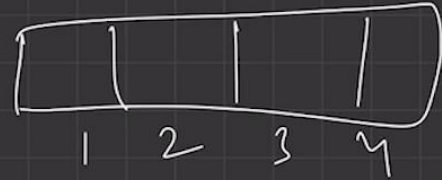
if No. of students $>$ no. of books



$m = 5$
 \rightarrow Return -1;



if No. of students $>$ no. of books



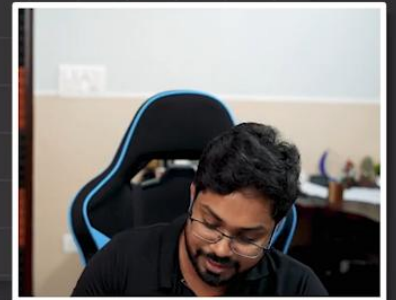
$$m = 5$$

↳ Return -1;

① Brute force

② Better Solution

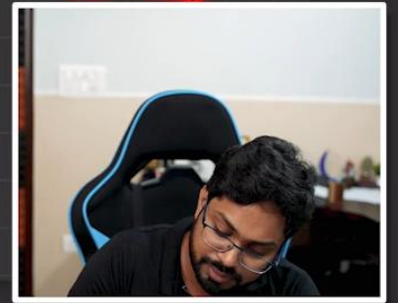
search space define



○
↓
min

202
↓
max

$M=1$
— 2
— 3
— 4
↓
~~5~~



○
↓
min



↓
2.2

203
↓
max



○
↓
min



202
↓
max.

2.?

① → $i = 1 \rightarrow 1?$

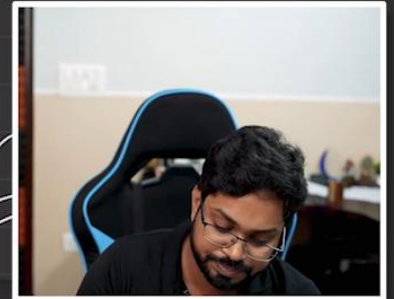
② → $i = 2 \rightarrow 2?$

③ → $i = 3 \rightarrow 3?$

—
|
|
|
|
113

→ $i = 113 \rightarrow 113?$

learn.thecodehelp.in
jhak6587@gmail.com
+919319347281



①

0 _____ 203

$$\text{int mid} = \frac{0+203}{2} = 101$$

Try to allocate each student at most (mid) 101 pages.

12 | 34 | 67 | 90

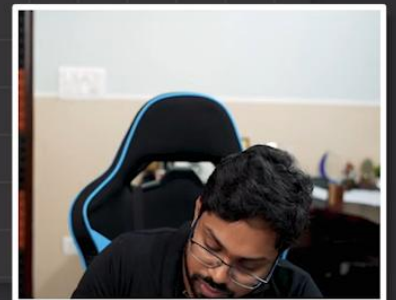
I $\rightarrow \underline{12+34} + \cancel{67} = \cancel{113}$

10

II $\rightarrow 67 + \cancel{90} = \cancel{157}$

X

III $\rightarrow \cancel{90} \rightarrow$



①

0 _____ 203

$$\text{int mid} = \frac{0 + 203}{2} = 101$$

learn.thecodehelp.in
jhak6587@gmail.com
+919319347281

Try to allocate each student at most
(mid) 101 pages.

12 | 34 | 67 | 90

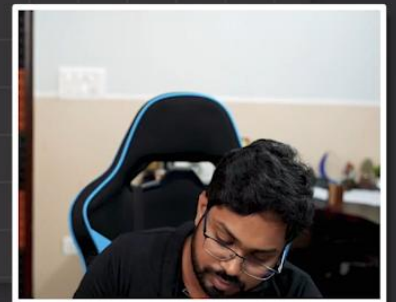
$$\text{I} \rightarrow \underline{12 + 34} + \cancel{67} = \cancel{113}$$

$$\text{II} \rightarrow 67 + \cancel{90} = \cancel{157}$$

$$\text{III} \rightarrow \cancel{90} \rightarrow$$

101 is Not a possible soln

X



①



$$\text{int mid} = \frac{0 + 203}{2} = 101$$

Try to allocate each student at most
(mid) 101 pages.

12 | 34 | 67 | 90

$$\text{I} \rightarrow \underline{12 + 34} + \cancel{67} = \cancel{113}$$

$$\text{II} \rightarrow 67 + \cancel{90} = \cancel{157}$$

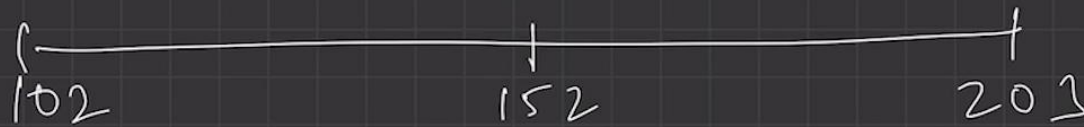
101 is Not a possible soln

if (Not possible)

{ start = mid + 1;
}



(2)



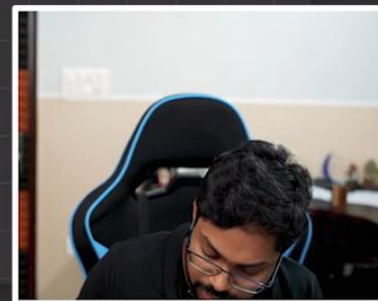
$$\text{mid} = \frac{102 + 203}{2} = 152.5 \rightarrow \text{at max pages,}$$

$$\text{I} \rightarrow 12 + \frac{24 + 67}{113} + 90 = 203$$

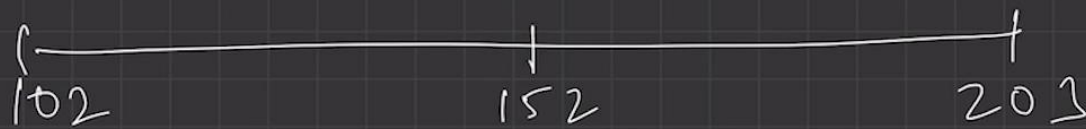
$$\text{II} \rightarrow 90$$

Possible solⁿ ✓

if



(2)



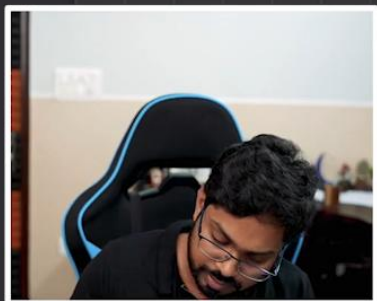
$$\text{mid} = \frac{102 + 203}{2} = 152.5 \rightarrow \text{at max pages,}$$

$$\text{I} \rightarrow 12 + \frac{24 + 67}{113} + 90 = 203$$

$$\text{II} \rightarrow 90$$

Possible solⁿ ✓

if (Possible solⁿ)
{
 // minimize pages.
 end = mid - 1;
}



(3)

$$102 \quad \text{---} \quad \text{---} \quad 51$$

126

I $\rightarrow 12 + 34 + 67 + 90 = 203$

learn.thecodehelp.in
jhak6587@gmail.com
+919319347281

II \rightarrow

90,

126 is also possible solⁿ



(3)

$$102 \quad \text{---} \quad \text{---} \quad \text{---} \quad 51$$

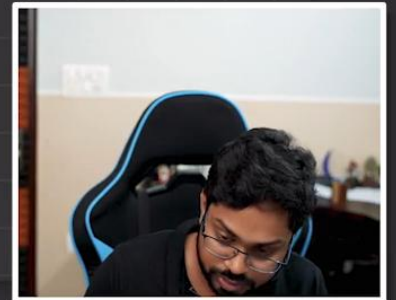
126

I $\rightarrow 12 + 34 + 67 + 90 = 203$

II $\rightarrow 90, \quad 126$ is also possible solⁿ

end = mid - 1

4



③

$$102 \quad \text{-----} \quad 151$$

126

I $\rightarrow 12 + 34 + 67 + 90 = 203$

II $\rightarrow 90$

226 is also possible solⁿ

end = mid - 1

④

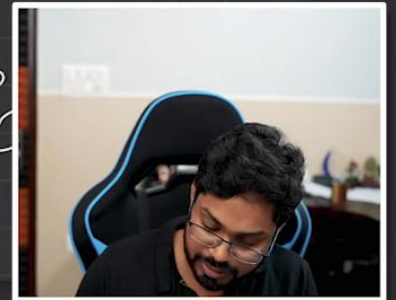
$$102 \quad \text{-----} \quad 125$$

113

I $\rightarrow 12 + 34 + 67 + 90 = 203$

II $\rightarrow 90$

Possible



(3)

$$102 \quad \text{-----} \quad 151$$

126

I $\rightarrow 12 + 34 + 67 + 90 = 203$

II $\rightarrow 90$, 226 is also possible solⁿ

$\text{end} = \text{mid} - 1$

(4)

$$102 \quad \text{-----} \quad 125$$

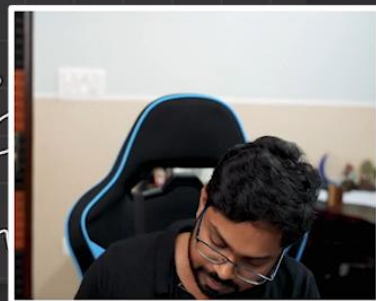
113

I $\rightarrow 12 + 34 + 67 + 90 = 203$

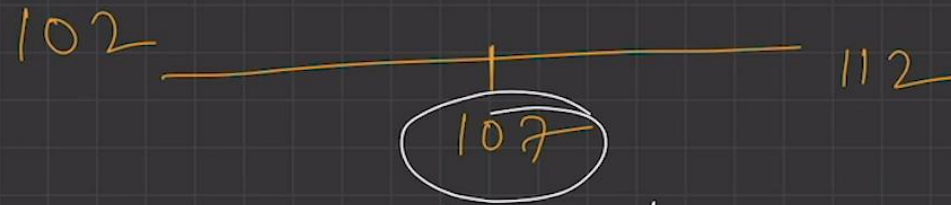
II $\rightarrow 90$

Possible

$\text{end} = \text{mid}$



5

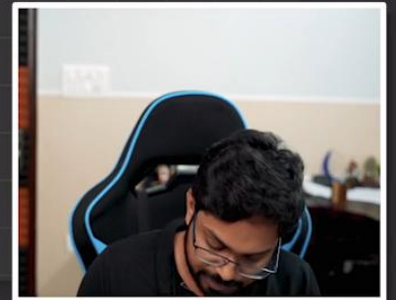


I $\rightarrow 12 + 34 + 67 = 113$

II $\rightarrow 67 + 90 = 157$

III $\rightarrow 90 \times$

Not possible solⁿ.
{ start = mid + 1
}



⑥

$$\begin{array}{ccc} 108 & & 112 \\ & \xrightarrow{\quad 110 \quad} & \end{array}$$

I \rightarrow

$$12 + 24 + \cancel{6/7} = 11/3$$

II \rightarrow

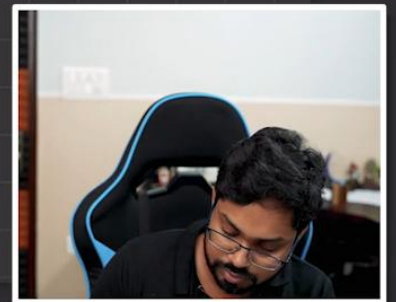
$$67 + \cancel{90} = \cancel{15/7}$$

III \rightarrow

~~90~~

$$\rightarrow M = 2.$$

Not pos.



$$\frac{1}{10} \rightarrow 112$$

$$6/7 = 11/3$$

$$6 = 15/7$$

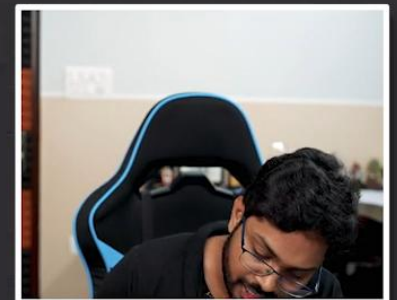
$$\rightarrow M = 2$$

not possible

$$\boxed{\text{start} = \text{mid} + 1}$$



Release to
Add Page



(7)

111 $\xrightarrow{\quad}$ 112
111 \rightarrow mid,

I \rightarrow

$$12 + 34 + 6/7 = 11/3$$

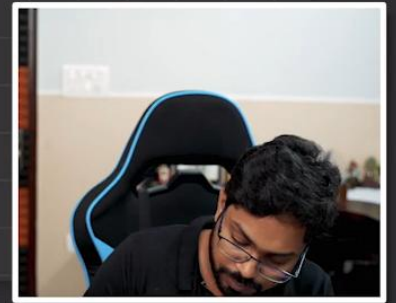
II \rightarrow

$$67 + 9/0 = 15/7$$

III \rightarrow

90 X, ,

learn.thecodehelp.in
jhak6587@gmail.com
+919319347281



(7)

111 $\xrightarrow{\quad}$ 112
111 \rightarrow mid,

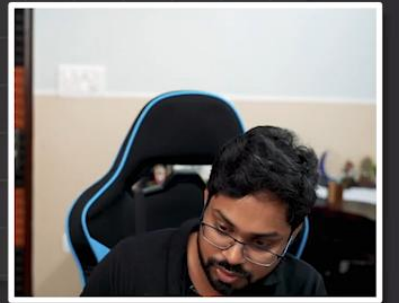
I \rightarrow $12 + 34 + 6/7 = 11/3$

II \rightarrow $67 + 9/0 = 15/7$

III \rightarrow 90 X.

Not possible.

\hookrightarrow start = mid + 1



8

112 ——— 112
112

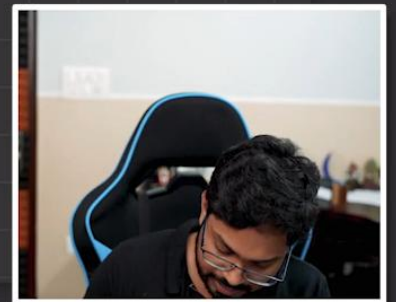
I → $12 + 34 + 67 = 113$

II → $67 + 90 = 157$

III → $90 \times$ M=2

Start = mid + 1

↳ 113,



⑧

112 ———— 112
 112

I → $12 + 34 + 67 = 113$

II → $67 + 90 = 157$

III → $90 \times$ M=2

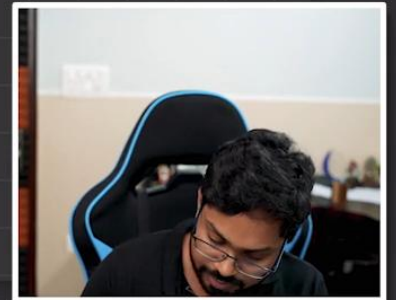
start = mid + 1

↳ 113

start = 113

end = 112

→ STOP



DSAWeek4HW - Google Drive x Allocate minimum number of p. x +

practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

Gmail YouTube Maps

Problems Courses Get Hired Contests </>POTD

Problem

Editorial

Submissions

Comments

code given n number of books. Every book has a number of pages. You have to allocate contiguous 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 the 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:

C++ (g++ 5.4)

Average Time: 35m

Start Timer

```
1 // } Driver Code Ends
9 //User function template in C++
10
11 class Solution
12 {
13 public:
14
15 bool isPossibleSolution(int A[], int N, int M, int sol){
16     int pageSum = 0;
17     int c = 1;
18     for(int i=0;i<N;i++){
19         if(A[i] > sol){
20             return false;
21         }
22         if(pageSum + A[i] > sol){
23             c++;
24             pageSum = A[i];
25         }
26     }
27 }
28
29 //Function to find minimum number of pages.
30 int findPages(int A[], int N, int M)
31 {
32     if(M>N) return -1;
33
34     int start = 0;
35     int end = accumulate(A, A+N, 0);
36     int ans = -1;
37
38     while(start <= end){
39         int mid = (start + end) >> 1;
40         if(isPossibleSolution(A, N, M, mid)){
41             ans = mid;
42             end = mid - 1;
43         }
44         else{
45             start = mid + 1;
46         }
47     }
48     return ans;
49 }
50 };
51 // } Driver Code Ends
```

130

DSAWeek4HW - Google Drive x Allocate minimum number of p. x +

practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

Gmail YouTube Maps

Problems Courses Get Hired Contests POTD

Problem Editorial Submissions Comments

Allocate minimum number of pages. Every book has a number of pages. You have to allocate contiguous 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 the 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:

ae

C++ (g++ 5.4) Average Time: 35m Start Timer

```
1 // } Driver Code Ends
9 //User function template in C++
10
11 class Solution
12 {
13 public:
14
15 bool isPossibleSolution(int A[], int N, int M, int sol){
16     int pageSum = 0;
17     int c = 1;
18     for(int i=0;i<N;i++){
19         if(A[i] > sol){
20             return false;
21         }
22         if(pageSum + A[i] > sol){
23             c++;
24             pageSum = A[i];
25             if(c > M){
26                 return false;
27             }
28         }
29         else{
30             pageSum += A[i];
31         }
32     }
33     return true;
34 }
35
36 //Function to find minimum number of pages.
37 int findPages(int A[], int N, int M)
38 {
39     if(M>N) return -1;
40
41     int start = 0;
42     int end = accumulate(A, A+N, 0);
43     int ans = -1;
44
45     while(start <= end){
46         int mid = (start + end) >> 1;
47         if(isPossibleSolution(A, N, M, mid)){
48             ans = mid;
49             end = mid - 1;
50         }
51         else{
52             start = mid + 1;
53         }
54     }
55     return ans;
56 }
```


DSAWeek4HW - Google DriveAllocate minimum number of p. x +

practice.geeksforgeeks.org/problems/allocate-minimum-number-of-pages0937/1?utm_source=gfg&utm_medium=article&utm_campaign=bottom_sticky_on_article

GmailYouTubeMaps

ProblemsCoursesGet HiredContestsPOTD

ProblemEditorialSubmissionsComments

Problem

Allocate minimum number of pages

You are given N number of books. Every book has a number of pages. You have to allocate contiguous 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 the 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:

GeeksforGeeks

C++ (g++ 5.4)Average Time: 35mStart Timer

```
16 int pageSum = 0;
17 int c = 1;
18 for(int i=0; i<N; i++){
19     if(A[i] > sol){
20         return false;
21     }
22     if(pageSum + A[i] > sol){
23         c++;
24         pageSum = A[i];
25         if(c > M){
26             return false;
27         }
28     }
29     else{
30         pageSum += A[i];
31     }
32 }
33 return true;
34 }
35
36 //Function to find minimum number of pages.
37 int findPages(int A[], int N, int M)
38 {
39     if(M>N) return -1;
40
41     int start = 0;
42     int end = accumulate(A, A+N, 0);
43     int ans = -1;
44
45     while(start <= end){
46         int mid = (start + end) >> 1;
47         if(isPossibleSolution(A, N, M, mid)){
48             ans = mid;
49             end = mid - 1;
50         }
51         else{
52             start = mid + 1;
53         }
54     }
55     return ans;
56 }
57 };
58 // } Driver Code Ends
```

