Data Structures and algo in Java - day 26

public class day26

{

public static void main(String[] args)

{

int arr [] = {25,46,28,49,24};

int students = 4;

allocateBooks(arr,students);

}

public static void allocateBooks(int arr [] , int students)

{

int low = getlow(arr);

int high = gethigh(arr);

int ans = -1;

while(low<=high)

{

int mid = (low+high)/2;

if(canAllocate(arr,students,mid)==true)

{

ans = mid;

high = mid -1;

}

else

{

low = mid +1;

}

}

System.out.println(ans+" is the minimum of the maximum books can be given to students");

}

public static int getlow(int arr[])

{

int low = Integer.MAX\_VALUE;

for(int i=0;i<arr.length;i++)

{

if(arr[i]<low)

{

low = arr[i];

}

}

return low;

}

public static int gethigh(int arr[])

{

int sum = 0;

for(int i=0;i<arr.length;i++)

{

sum = sum + arr[i];

}

return sum;

}

public static boolean canAllocate(int arr[], int students, int limit)

{

int studentCount = 1;

int currentSum = arr[0];

for(int i=1;i<arr.length;i++)

{

if(arr[i]+currentSum>limit)

{

studentCount = studentCount +1;

currentSum = arr[i];

}

else

{

currentSum = currentSum+arr[i];

}

}

if(studentCount <= students)

{

return true;

}

return false;

}

}