Data structures and algo in Java - Day 24

public class day24

{

public static void main (String [] args)

{

// // ----------------------------

// int arr [] = {7,7,7,7,13,11,12,7};

// int m = 2;// no of bouquets

// int k = 3;// adjacent flowers

// int ans = minNoOfDaysToMakeABouquet(arr,m,k);

// if(ans == -1)

// {

// System.out.println("We cant Make Bouquets");

// }

// else{

// System.out.println("The minimum No of Days to Make "+m+" no of Bouquets with "+k+" Adjacent flowers is "+ans);

// }

// // -----------------------------

// // -----------------------------

// int arr [] = {1,2,5,9};

// int t = 7;

// int ans = smallestDivisorWithThreshold(arr,t);

// // -----------------------------

int arr [] = {1,2,3,4,5,6,7,8,9,10};

int days = 5;

capacityOfShip(arr, days);

}

public static int minNoOfDaysToMakeABouquet(int arr[], int m, int k)

{

int low = getLow(arr);

int high = getHigh(arr);

int ans = 0;

if(arr.length < m\*k)

{

return -1;

}

while(low<=high)

{

int mid = (low+high)/2;

if(possible(arr,mid,m,k)==true)

{

ans = mid;

high = mid -1;

}

else

{

low = mid +1;

}

}

return ans;

}

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 high = Integer.MIN\_VALUE;

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

{

if(arr[i]>high)

{

high = arr[i];

}

}

return high;

}

public static boolean possible(int arr[], int day , int m , int k)

{

int count = 0;

int noOfBouquets = 0;

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

{

if(arr[i]<=day) // 7<=7

{

count++;

}

else

{

noOfBouquets = noOfBouquets + (count/k);

count = 0;

}

}

noOfBouquets = noOfBouquets + (count/k);

if(noOfBouquets >=m)

{

return true;

}

return false;

}

public static int smallestDivisorWithThreshold(int arr[], int threshold)

{

int low = 1;

int high = getHigh(arr);

int ans = 0;

while(low<=high)

{

int mid = (low+high)/2;

if(acceptable(arr,mid,threshold)==true)

{

ans = mid;

high = mid-1;

}

else{

low = mid +1;

}

}

return ans;

}

public static boolean acceptable(int arr[], int mid , int threshold)

{

int sum = 0;

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

{

sum = sum + (int) Math.ceil((double)arr[i]/mid);

}

if(sum>threshold)

{

return false;

}

return true;

}

public static void capacityOfShip(int arr[], int days)

{

int low = getHigh(arr);

int high = getSum(arr);

int ans = 0;

while(low<=high)

{

int mid = (low+high)/2;

int daysRequired = findDays(arr,mid);

if(daysRequired<=days)

{

ans = mid;

high = mid -1;

}

else

{

low = mid +1;

}

}

System.out.println("The minimum capacity required to fill the ship in "+days+" days is "+ans);

}

public static int findDays(int arr[], int capacity)

{

int day = 1;

int load = 0;

for(int i=0;i<arr.length;i++) // 1,2,3,4,5,6,7,8,9,10

{

if(load+arr[i]>capacity)//32

{

day = day + 1;

load = arr[i]; // next day load

}

else

{

load = load + arr[i]; // current day load

}

}

return day;

}

public static int getSum(int arr[])

{

int sum = 0;

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

{

sum = sum + arr[i];

}

return sum;

}

}