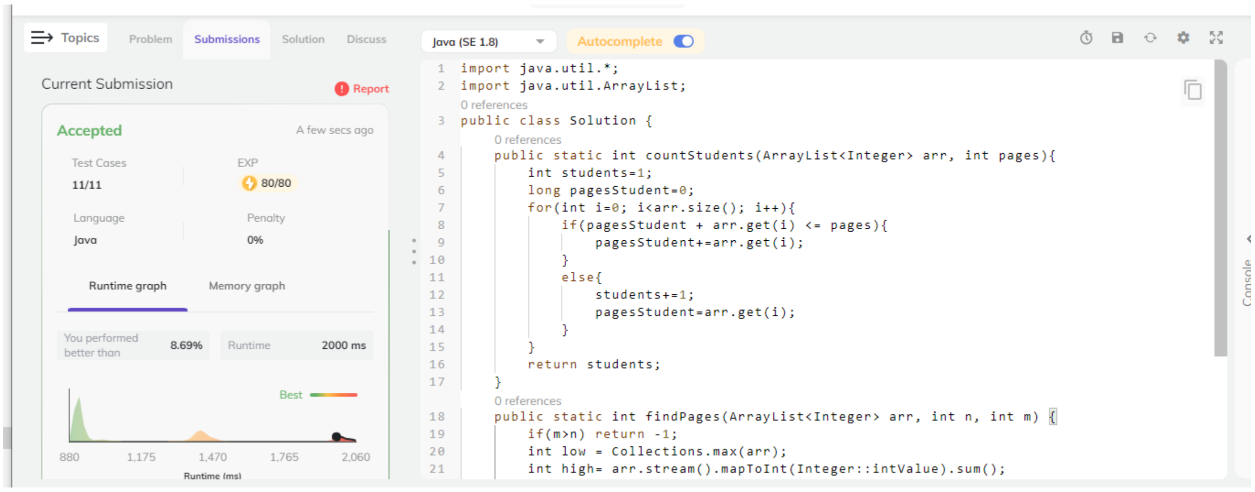
Classwork

Allocate Books [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/allocate-books_1090540?leftPanelTab=0



Current Submission
Report

Accepted
A few secs ago

Test Cases 11/11
EXP 80/80
Language Java
Penalty 0%

You performed better than 8.69%
Runtime 2000 ms

```
import java.util.*;
import java.util.ArrayList;
0 references
public class Solution {
    0 references
    public static int countStudents(ArrayList<Integer> arr, int pages){
        int students=1;
        long pagesStudent=0;
        for(int i=0; i<arr.size(); i++){
            if(pagesStudent + arr.get(i) <= pages){
                pagesStudent+=arr.get(i);
            }
            else{
                students+=1;
                pagesStudent=arr.get(i);
            }
        }
        return students;
    }
    0 references
    public static int findPages(ArrayList<Integer> arr, int n, int m) {
        if(m>n) return -1;
        int low = Collections.max(arr);
        int high= arr.stream().mapToInt(Integer::intValue).sum();
```

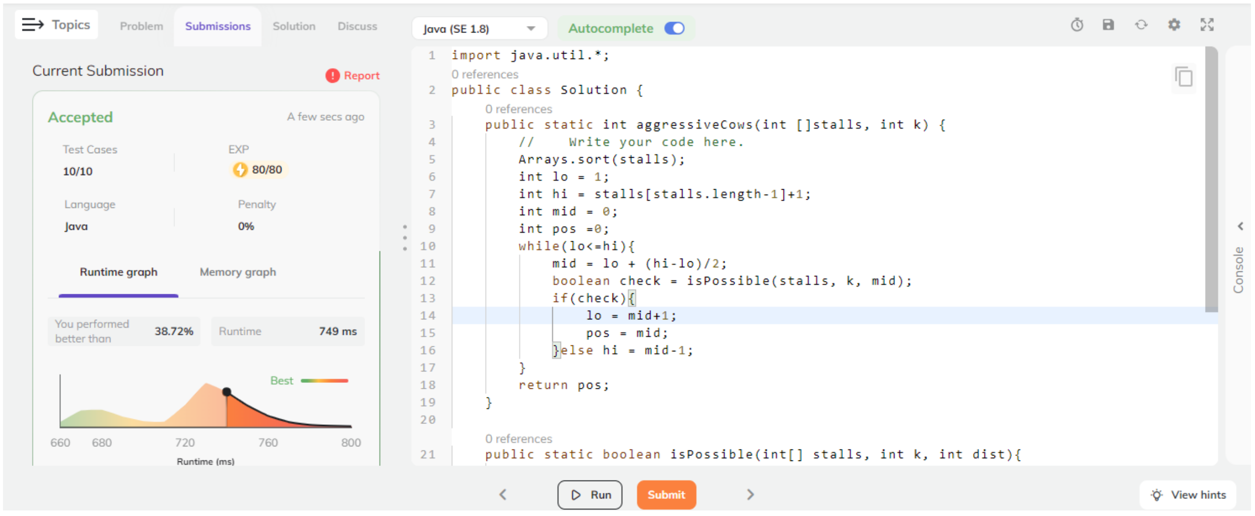
Java Code

```
import java.util.*;
import java.util.ArrayList;
public class Solution {
    public static int countStudents(ArrayList<Integer> arr, int pages){
        int students=1;
        long pagesStudent=0;
        for(int i=0; i<arr.size(); i++){
            if(pagesStudent + arr.get(i) <= pages){
                pagesStudent+=arr.get(i);
            }
            else{
                students+=1;
                pagesStudent=arr.get(i);
            }
        }
        return students;
    }
    public static int findPages(ArrayList<Integer> arr, int n, int m) {
        if(m>n) return -1;
        int low = Collections.max(arr);
        int high= arr.stream().mapToInt(Integer::intValue).sum();
        while(low<=high){
            int mid=(low+high)/2;
            int students=countStudents(arr, mid);
            if(students > m) low=mid+1;
            else high = mid-1;
        }
        return low;
    }

}
```

Aggressive Cows [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/aggressive-cows_1082559?source=youtube&campaign=love_babbar_codestudio2&utm_source=youtube&utm_medium=affiliate&utm_campaign=love_babbar_codestudio2&leftPanelTab=0



Current Submission
Report

Accepted
A few secs ago

Test Cases 10/10
EXP 80/80
Language Java
Penalty 0%

You performed better than 38.72%
Runtime 749 ms

```
import java.util.*;
0 references
public class Solution {
    0 references
    public static int aggressiveCows(int []stalls, int k) {
        //    Write your code here.
        Arrays.sort(stalls);
        int lo = 1;
        int hi = stalls[stalls.length-1]+1;
        int mid = 0;
        int pos =0;
        while(lo<=hi){
            mid = lo + (hi-lo)/2;
            boolean check = isPossible(stalls, k, mid);
            if(check){
                lo = mid+1;
                pos = mid;
            }else hi = mid-1;
        }
        return pos;
    }

    0 references
    public static boolean isPossible(int[] stalls, int k, int dist){
```

Java Code

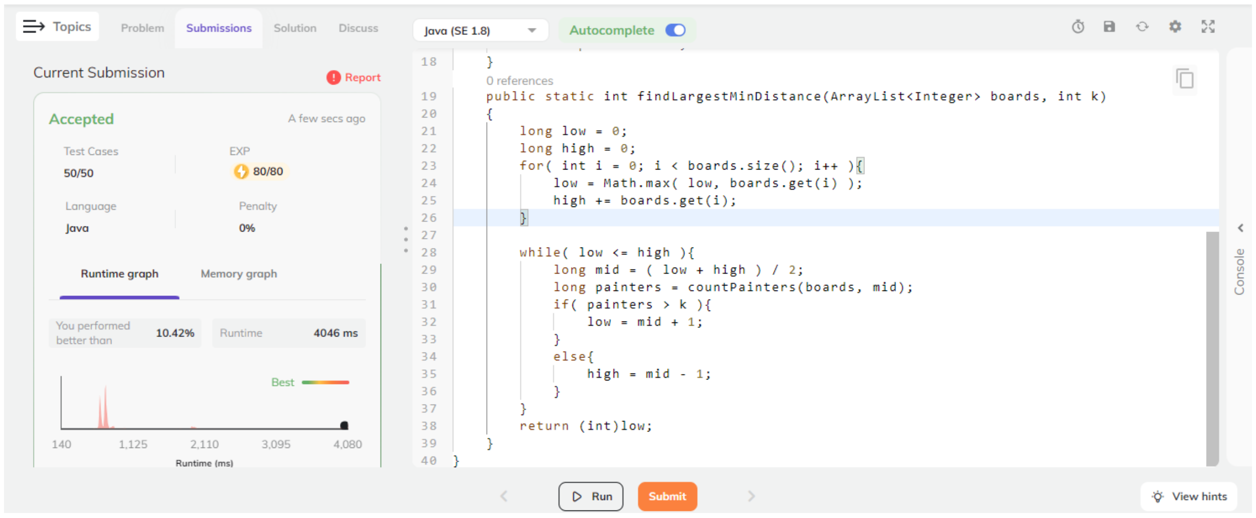
```
import java.util.*;
public class Solution {
    public static int aggressiveCows(int []stalls, int k) {
        //    Write your code here.
        Arrays.sort(stalls);
        int lo = 1;
        int hi = stalls[stalls.length-1]+1;
        int mid = 0;
        int pos =0;
        while(lo<=hi){
            mid = lo + (hi-lo)/2;
            boolean check = isPossible(stalls, k, mid);
            if(check){
                lo = mid+1;
                pos = mid;
            }else hi = mid-1;
        }
        return pos;
    }

    public static boolean isPossible(int[] stalls, int k, int dist){
        k--;
        int curr = 0;
        int i=1;
        while(k>0&&i<stalls.length){
            if(stalls[i]-stalls[curr] >=dist){
                k--;
                curr = i;
            }
            i++;
        }
        if(k>0) return false;
        return true;
    }
}
```

Homework

Painter's Partition Problem [Coding Ninjas]

Question Link:-
https://www.codingninjas.com/studio/problems/painter's-partition-problem_1089557?source=youtube&campaign=love_babbar_codestudio2&utm_source=youtube&utm_medium=affiliate&utm_campaign=love_babbar_codestudio2&leftPanelTab=0



Current Submission

Accepted A few secs ago

Test Cases 50/50
EXP 80/80
Language Java
Penalty 0%

You performed better than 10.42%
Runtime 4046 ms

```java
    }
    0 references
    public static int findLargestMinDistance(ArrayList<Integer> boards, int k)
    {
        long low = 0;
        long high = 0;
        for( int i = 0; i < boards.size(); i++ ){
            low = Math.max( low, boards.get(i) );
            high += boards.get(i);
        }

        while( low <= high ){
            long mid = ( low + high ) / 2;
            long painters = countPainters(boards, mid);
            if( painters > k ){
                low = mid + 1;
            }
            else{
                high = mid - 1;
            }
        }
        return (int)low;
    }
}
```

Java Code

```java
import java.util.ArrayList;
import java.lang.Math;
public class Solution
{
    static long countPainters( ArrayList<Integer> arr, long max ){
        long currBoards = 0;
        long painterCount = 1;
        for( int i = 0 ; i < arr.size() ; i++ ){
            if( currBoards + arr.get(i) <= max ){
                currBoards += arr.get(i);
            }
            else{
                painterCount++;
                currBoards = arr.get(i);
            }
        }
        return painterCount;
    }
    public static int findLargestMinDistance(ArrayList<Integer> boards,
int k)
    {
        long low = 0;
        long high = 0;
        for( int i = 0; i < boards.size(); i++ ){
            low = Math.max( low, boards.get(i) );
            high += boards.get(i);
        }

        while( low <= high ){
            long mid = ( low + high ) / 2;
            long painters = countPainters(boards, mid);
            if( painters > k ){
                low = mid + 1;
            }
            else{
                high = mid - 1;
            }
        }
        return (int)low;
    }
}
```