

The painter

$n = 4 \rightarrow$ no of boards
 $\begin{array}{cccc} 0 & 1 & 2 & 3 \end{array}$

$$\text{Arr} = [10, 10, 10, 10]$$

$K = 2 \rightarrow$ no of painters

Find minimum time to paint all board

Condition:- Painter can paint continuous section of board

Painter 1 $\rightarrow [1, 2, 3], [2, 3], [0, 1, 2, 3], [0] \checkmark$
 $[1, 3], [0, 2], [0, 1, 3] \times$

$K = 2$.

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ [10, 10, 10, 10] \end{array}$$

$$10 + 10 \rightarrow \text{Painter 1} = 20$$

$$\frac{2}{10} + \frac{3}{10} \rightarrow \text{Painter 2} = 20$$

Minimum time required = 20

$$\begin{array}{cccc} 0 & 1 & 2 & 3 \\ [20, 30, 40, 80] \end{array}$$

$$1 \left\{ \begin{array}{l} 1st \rightarrow 20 + 30 = 50 \\ 2nd \rightarrow 40 + 80 = 120 \\ \text{minimum time required} \rightarrow \underline{120} \end{array} \right.$$

$$2 \left\{ \begin{array}{l} 1st \rightarrow 20 + 30 + 40 = 90 \\ 2nd \rightarrow 80 \rightarrow 80 \\ \text{minimum time required} \rightarrow \underline{90} \end{array} \right. \checkmark$$

$\backslash 2nd \rightarrow 00 \rightarrow 0$
 minimum time required $\rightarrow \underline{90}$

$$3 \left\{ \begin{array}{l} 1st \rightarrow 20 = 20 \\ 2nd \rightarrow 30 + 40 + 80 = 150 \\ \text{minimum time} = 150 \end{array} \right.$$

Answer should be 90.

Solution:-

[20, 30, 40, 80]

$$\text{maxTime consumed by painter} = 20 + 30 + 40 + 80 \\ = 170$$

minTime consumed by painter to paint all the
 [20, 30, 40, 80] board = board with maximum
 length $= \underline{80}$.

$$\text{time} \rightarrow 80 \quad 170 \\ \underline{\text{min}} \quad \underline{\text{max.}}$$

$$\underline{k=2.}$$

$$\underline{\text{left}} = \text{min}$$

$$\underline{\text{right}} = \text{max}$$

$$\text{mid} = (\text{left} + \text{right}) / 2 = (80 + 170) / 2 = 250 / 2 \\ = \underline{125}$$

int left = min; (maximum value of array)

int right = max; (sum of all values of array)
 int ans = -1;

while (left <= right) {

int mid = (left + right) / 2;

```

int mid = (left + right) / 2;
if (canPaintAllBoard(boards, mid, K)) {
    ans = mid;
    right = mid - 1;
} else {
    left = mid + 1;
}

```

$\begin{bmatrix} 20, 30, 40, 80 \\ \underline{\hspace{2cm}} \end{bmatrix}$
~~80~~

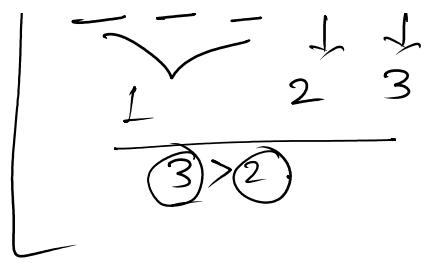
```

public boolean canPaintAllBoard(int boards[], int time, int k) {
    int totalTime = 0; int noOfPainter = 1;
    for (int i = 0; i < boards.length; i++) {
        totalTime += boards[i];
        if (totalTime > time) {
            noOfPainter++;
            totalTime = boards[i];
        }
    }
    if (noOfPainter <= k) {
        return true;
    }
    return false;
}

```

Test Case 1.
 $\begin{cases} \text{time} \rightarrow 125 \\ k \rightarrow 2 \end{cases}$
board
 $\begin{bmatrix} 20, 30, 40, \underline{80} \\ \underline{\hspace{2cm}} \end{bmatrix}$
 1 2

Test Case 2.
 $\begin{cases} \text{time} = 125 \\ k = 2 \\ \text{board} \end{cases}$
 $\begin{bmatrix} 20, 30, 40, 80, 90 \\ \underline{\hspace{2cm}} \end{bmatrix}$
 1 2 3



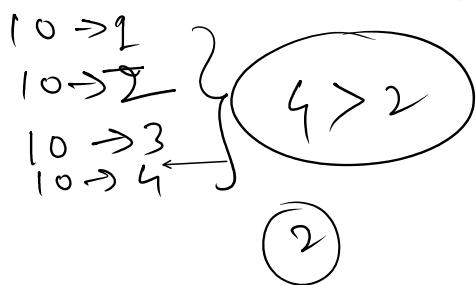
$[10, 10, \underline{10}, 10] \rightarrow k=2$

$$\min = 10$$

$$\max = 25$$

$$\text{mid} = 25$$

CanPaintAllBoard(board, 25, 2)



$$\begin{aligned} & 10 \downarrow 25 \\ & (10+25)/2 \\ & = 35/2 \\ & = 17 \\ & 18 \quad 24 \\ & \textcircled{21} \end{aligned}$$

```

4 public class Solution {
5
6     public static void main(String[] args) {
7         /* Enter your code here. Read input from STDIN. Print output to
8          * STDOUT. Your class must be named Solution.
9          */
10        Scanner sc = new Scanner(System.in);
11        int n = sc.nextInt();
12        int arr[] = new int[n];
13        for(int i=0;i<n;i++){
14            arr[i] = sc.nextInt();
15        }
16        int k = sc.nextInt(); // no of painters
17        int maxValue = Integer.MIN_VALUE;
18        for(int i=0;i<n;i++){
19            maxValue = Math.max(maxValue,arr[i]);
20        }
21        int left = maxValue;
22        int totallength=0;
23        for(int i=0;i<n;i++){
24            totallength+=arr[i];
25        }
26        int right = totallength;
27        int ans=-1;
28        while(left<=right){
29            int mid = (left+right)/2;
30            if(canPaint(arr,mid,k)){
31                ans = mid;
32            }else{
33                left = mid+1;
34            }
35        }
36    }

```

```
37     public static boolean canPaint(int arr[] , int time, int k){  
38         int npainter=1;  
39         int timeconsumed =0;  
40         for(int i=0;i<arr.length;i++){  
41             timeconsumed +=arr[i];  
42             if(timeconsumed>time){  
43                 npainter++;  
44                 timeconsumed = arr[i];  
45             }  
46         }  
47         if(npainter<=k){  
48             return true;  
49         }  
50         return false;  
51     }  
52 }
```

ArrayList

↳ Dynamic size array

In array, the size or length is fixed, therefore we have ArrayList which has dynamic length.

ArrayList is a class in Java, therefore it has built-in methods.

Syntax wrapper class of data type
 →

ArrayList<Type> name-of-arraylist
 = new ArrayList<>();

Example:-

ArrayList<Integer> myList = new ArrayList<>();

Built-in Methods

(i.) add(Element) → We can pass element to be added to the ArrayList

myList.add(5);

[⁰
5]

myList.add[6]; [⁰
5, ¹
6]

myList.add[10]; [⁰
5, ¹
6, ²
10]

(ii.) add (int index, T element)

↳ we can add element at specific index using this method.

myList.add(1, 20)

[^{0 1 2 3}
[5 20 6 10]]

(iii.) get (int index)

myList.get(3) → 10.

myList.get(0) → 5

(iv.) remove (int index) → Removes element at given index

myList.remove(0);

(v.) size() : — Size of arraylist or no of elements in the arraylist

(vi.) set (int index, T element)

↳ This will set element at the given index.

(vii.) contains (T element) : → It will return true or false, true → if element is present otherwise false.