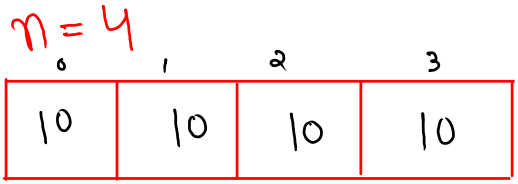
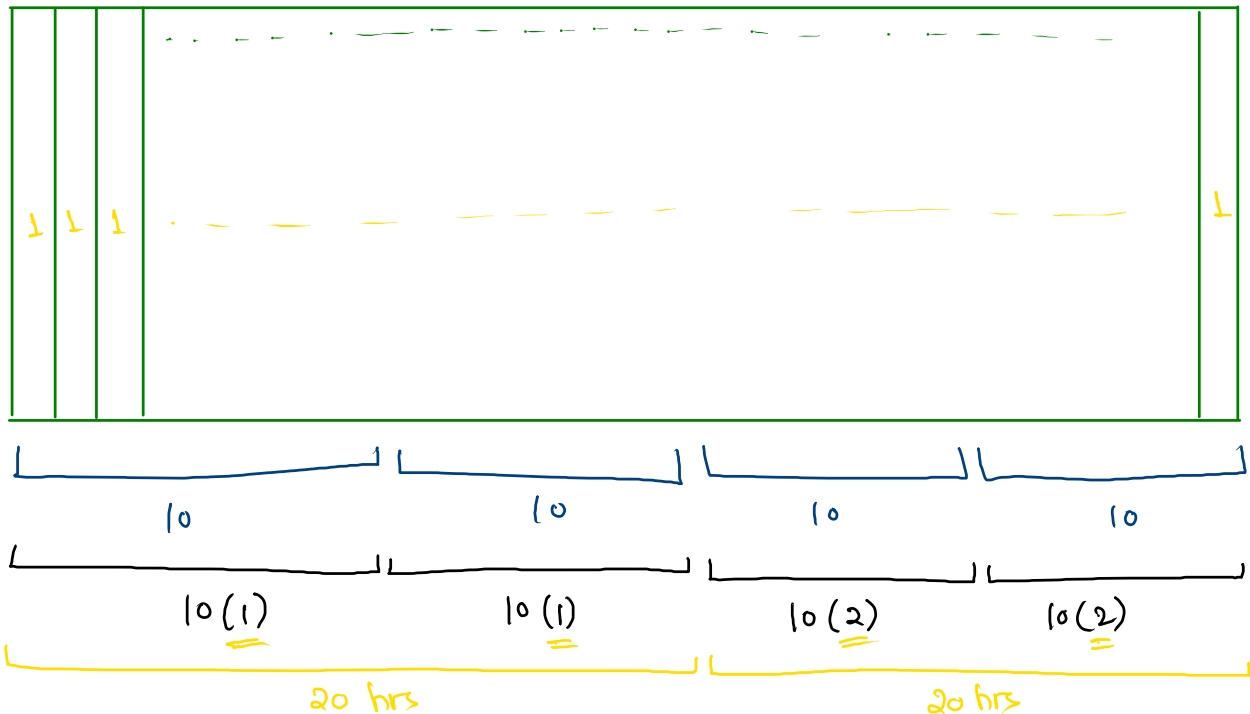
The painter



K = 2 // painters

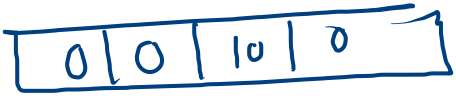


find range of "time"

- ✓ min = max of array elements
- ✓ max = sum of array elements

---

ex



min = 10
(max of array)

1)

```
public static void painter(int[] arr, int n, int k) {
    int mini = max( arr );
    int maxi = sum( arr );

    int si = mini;
    int ei = maxi;
    int ans = -1;
    while ( si <= ei ) {
        int mid = (si + ei) / 2;    // time
        if ( check(arr, mid) <= k ) {   // no of painters to paint all boards
            ans = mid;
            ei = mid - 1;   // decrease time
        } else {
            si = mid + 1;   // increase time
        }
    }
    System.out.println(ans);
}
```

si

2)

```
public static int check(int[] arr, int limit) {
    int sum = 0;
    int nop = 1;     // no of painters
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
        if (sum > limit) {
            nop++;
            sum = arr[i];
        }
    }
    return nop;
}
```

3)

```
public static int sum(int[] arr) {
    int ans = 0;
    for (int i = 0; i < arr.length; i++) {
        ans += arr[i];
    }
    return ans;
}
```

4)

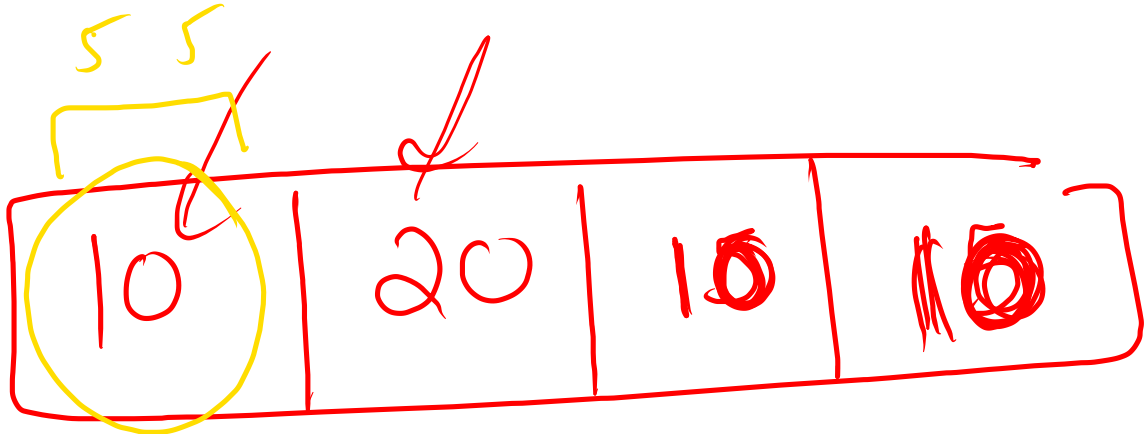
```
public static int max(int[] arr) {
    int ans = 0;
    for (int i = 0; i < arr.length; i++) {
        ans = Math.max( ans, arr[i] );
    }
    return ans;
}
```

Note:-

check fn is calculating no. of painters to paint all boards.

if no. of painters reqd to paint of all boards is more than k, means our time less

10 10 10 10



K=2

$\Rightarrow 50/2 = 25$

1 → 25

2 → 25

⇒ Arraylist (AL)

↳ dynamic array

Array → size was fixed
↳ it store premitive data type (int, float, ...)

(java)
Arraylist → size can be changed at any point of time
↳ it can store only Objects (Integer, Boolean, String, Double, Character ...) (wrapper classes)

C++
(Vector)

Inbuilt functions of AL

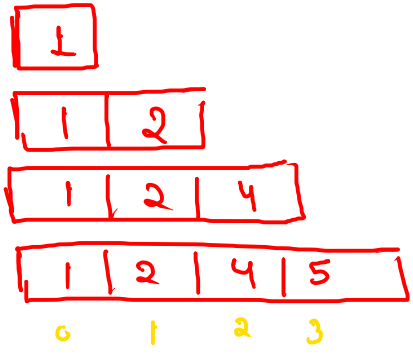
- add element in AL
- get element from AL
- remove/delete element from AL
- update/set element
- Iterate over AL

1) add elements in AL

```java
public static void main(String[] args) {
    ArrayList<Integer> arr = new ArrayList<>();

    // add elements in AL
    arr.add(1);
    arr.add(2);
    arr.add(4);
    arr.add(5);
}
```

AL of size 0



Note:- arr.add(val);

// it will add val in last of AL

2) how to get data from AL using .get() function;

```
public static void main(String[] args) {
    ArrayList<Integer> arr = new ArrayList<>();

    // add elements in AL
    arr.add(1);
    arr.add(2);
    for (int i = 0; i < arr.size(); i++) {
        System.out.print( arr.get(i) + " " );
    }
    System.out.println();

    arr.add(4);
    arr.add(5);
    for (int i = 0; i < arr.size(); i++) {
        System.out.print( arr.get(i) + " " );
    }
}
```

Syntax:-

arr_name.get(index)

3) how to find length of AL

arr_name.size();

4) how to remove an element

```
arr.remove(2);
for (int i = 0; i < arr.size(); i++) {
    System.out.print( arr.get(i) + " " );
}
System.out.println();
```

syntex:-

arr_name . remove (index) ;

5) how to set an element in AL

```
arr.set( 2, 3 );
for (int i = 0; i < arr.size(); i++) {
    System.out.print( arr.get(i) + " " );
}
System.out.println();
```

Syntax:-

arr_name.set(index, value);

6) how to add value in b/w to AL

```
arr.add( 2, 6 );
for (int i = 0; i < arr.size(); i++) {
    System.out.print( arr.get(i) + " " );
}
System.out.println();
```

Syntax:-

arr_name.add (index, value);

7) Sorting in AL

array ⟶ Arrays.sort(arr);

arraylist ⟶ Collections.sort(arr);

⇒ code for all

```java
public static void main(String[] args) {
    ArrayList<Integer> arr = new ArrayList<>(); // declare arraylist
    arr.add(1); // to add elemets
    arr.add(2);
    arr.add(4);
    arr.add(5);
    arr.remove(2);  // to remove element
    arr.add(5);
    arr.add(4);
    arr.set( 2, 3 );    // to update element
    arr.add( 2, 6 );
    Collections.sort(arr);  // to sort arraylist
    for (int i = 0; i < arr.size(); i++) {  // iterate over arraylist
        System.out.print( arr.get(i) + " " );
    }
    System.out.println();
}
```

← to add to some index

to get element

dry run of painter question

```
public static void painter(int[] arr, int n, int k) {
    int mini = max( arr );
    int maxi = sum( arr );

    int si = mini;
    int ei = maxi;
    int ans = -1;
    while ( si <= ei ) {
        int mid = (si + ei) / 2;     // time
        if ( check(arr, mid) <= k ) {   // no of painters to paint all boards
            ans = mid;
            ei = mid - 1;   // decrease time
        } else {
            si = mid + 1;   // increase time
        }
    }
    System.out.println(ans);
}
```

mini → 10
maxi → 40

② , 4

si

```
4
10 10 10 10
2
```

K = 2

25 → ② <=2 True
17 → 4 <=2 false
21 → ② <=2 True
19 → 4 <=2 False
20 → ② <=2 True

... 18, 19, 20, 21, 22, 23, 24 ...

ei

si

mid


```java
public static int check(int[] arr, int limit) {
    int sum = 0;
    int nop = 1;     // no of painters
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
        if (sum > limit) {
            nop++;
            sum = arr[i];
        }
    }
    return nop;
}
```

3

1

limit = 16 (total time given to us to paint all boards)

sum = 0 // total time for each painter

nop = ~~1~~ ~~2~~ 3

sum = 10,	10 > 16	False
sum = 35,	35 > 16	True
sum = 30	30 > 16	True
sum = 15	15 > 16	False