

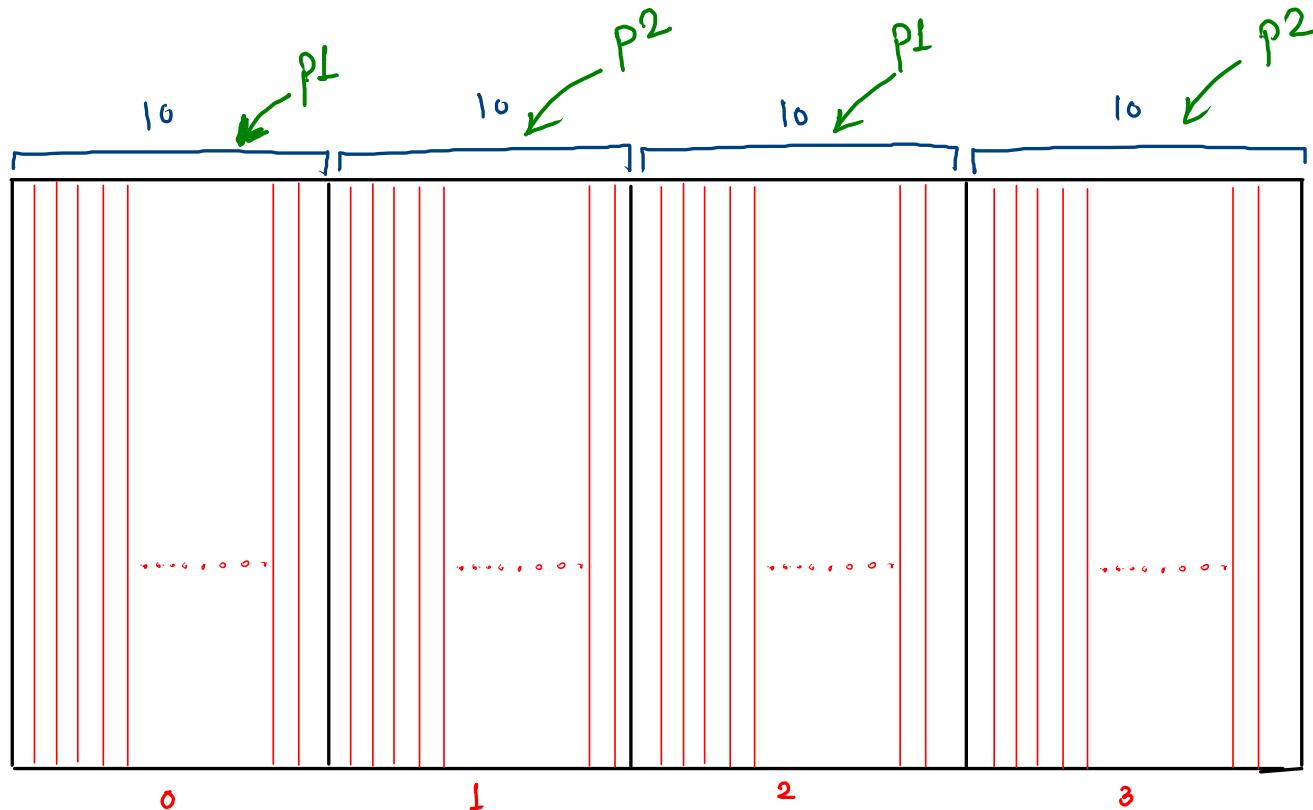
## The painter

$$n=4$$

$$\text{arr} = [ \underset{0}{10}, \underset{1}{10}, \underset{2}{10}, \underset{3}{10} ]$$

$$K=2$$

$$\underline{\underline{\text{Time} = 10 + 10 = 20}}$$



Note:- Only 1 painter will paint entire sequence of board

Ex

$n = 4$

$$\text{arr} = [10, 20, 30, 40], \underline{\underline{k=4}}$$

$p^1$      $p^2$      $p^3$      $p^4$

time = 40

$s_i \Rightarrow \max(\text{arr})$

Ex

$n = 4$

$$\text{arr} = [10, 20, 30, 40], \underline{\underline{k=1}}$$

$p^1$      $p^1$      $p^1$      $p^1$

time = 100

$$\text{time} = 10 + 20 + 30 + 40 = \underline{\underline{100}}$$

$e^i = \sum(\text{arr})$

Ex 1

$$arr = [ \underset{6}{10}, \underset{1}{10}, \underset{2}{10}, \underset{3}{10} ] \quad s_i = 10, e_i = 40$$

K = 2

$$K = 10 \dots \underset{17}{\underline{17}} \underset{18}{\underline{18}} \underset{19}{\underline{19}} \underset{20}{\underline{20}} \underset{21}{\underline{21}} \dots \underset{24}{\underline{24}} \underset{25}{\underline{25}} \dots \dots \dots 40$$

↑  
 e<sub>i</sub>  
 s<sub>i</sub>  
 ↑  
 mid

how many painters will  
take to finish painting  
all boards in  
mid time

$$arr = [ \underset{6}{10}, \underset{1}{10}, \underset{2}{10}, \underset{3}{10} ]$$

$$\underline{\underline{mid = 25}} \longrightarrow p_1(25), p_1(15), p_2(25), p_2(15) \quad \underline{\underline{P = 2}}$$

$$\underline{\underline{mid = 17}} \longrightarrow p_1(17), p_2(17), p_3(17), p_4(17) \quad \underline{\underline{P = 4}}$$

$$\underline{\underline{mid = 21}} \longrightarrow p_1(21), p_1(11), p_2(21), p_2(11) \quad \underline{\underline{P = 2}}$$

$$\underline{\underline{mid = 19}} \longrightarrow p_1(19), p_2(19), p_3(19), p_4(19) \quad \underline{\underline{P = 4}}$$

$$\underline{\underline{mid = 20}} \longrightarrow p_1(20), p_1(10), p_2(20), p_2(10) \quad \underline{\underline{P = 2}}$$

Code

```
1) public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int painters = scn.nextInt();
    int anns = paintersProblem(arr, n, painters);
    System.out.println(anns);
}

2) public static int paintersProblem(int[] arr, int n, int painters) {
    int si = max(arr);
    int ei = sum(arr);
    while ( si <= ei ) {
        int mid = (si + ei) / 2; // mid is time
        if (check(arr, mid) > painters) { // how many painters will I take to painte all in mid time
            si = mid + 1;
        } else {
            ei = mid - 1;
        }
    }
    return si;
}

3) public static int check(int[] arr, int time) {
    int painter = 1;
    int sum = 0;
    for (int i = 0; i < arr.length; i++) {
        sum += arr[i];
        if (sum > time) {
            painter++;
            sum = arr[i];
        }
    }
    return painter;
}

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;
}

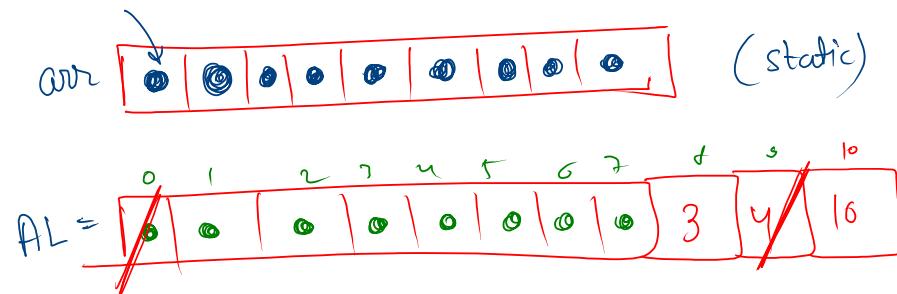
5) public static int sum(int[] arr) {
    int ans = 0;
    for (int i = 0; i < arr.length; i++) {
        ans += arr[i];
    }
    return ans;
}
```

$$T_0 C = \underline{\underline{O(2n + n \log n)}}$$

# ArrayList (dynamic in nature)

Note :-

AL can store only  
Objects



→ Integer, Character, Boolean, String etc...  
(Wrapper)

Syntax

ArrayList<DataType> arr = new ArrayList<>();

Ex:-

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

→ size = 0

size of AL will be zero at the time of declaration.

## Inbuilt function

1) arr.size() // 0

2) arr.isEmpty() // true

### 3) add elements in AL

arr.add(value); // value will be added  
on the rightmost side of  
AL

arr.add(index, value); // value will be added at index

```
public static void main(String[] args) {
```

```
    ArrayList<Integer> arr = new ArrayList<>(); → size = 0
```

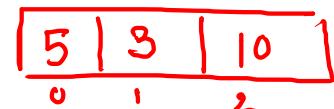
```
    arr.add(5); → [5]
```

```
    arr.add(3); → [5 | 3]
```

```
    arr.add(10); → [5 | 3 | 10]
```

```
    System.out.println(arr.size());
```

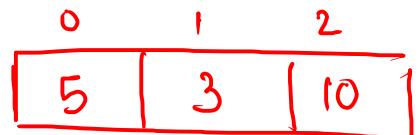
```
}
```



4) how to access any element from AL

↳ arr.get(index);

```
public static void main(String[] args) {  
    ArrayList<Integer> arr = new ArrayList<>();  
    arr.add(5);  
    arr.add(3);  
    arr.add(10);  
    System.out.println(arr.size());  
    System.out.println(arr.get(2));  
}
```

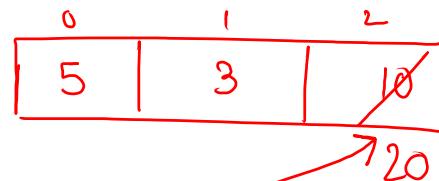


5) how to update any element in AL

→ arr[index] = value; (array)

→ arr.set(index, value); (AL)

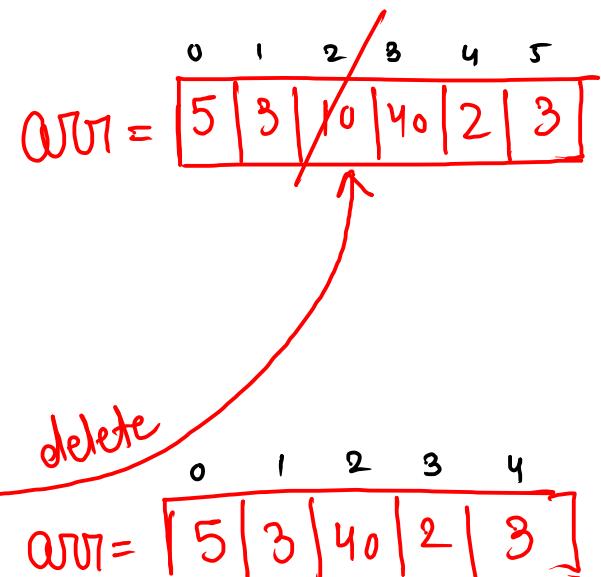
```
public static void main(String[] args) {  
    ArrayList<Integer> arr = new ArrayList<>();  
    arr.add(5);  
    arr.add(3);  
    arr.add(10);  
    System.out.println(arr.size());  
    arr.set(2, 20);  
    System.out.println(arr.get(2));  
}
```



6) how to delete any element from AL

→ `ArrayList.remove(index);`

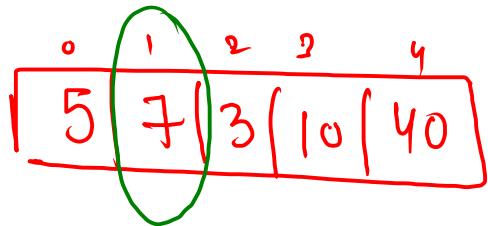
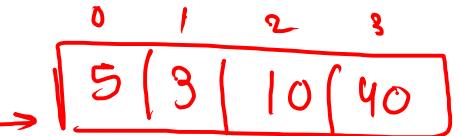
```
public static void main(String[] args) {  
    ArrayList<Integer> arr = new ArrayList<>();  
    arr.add(5);  
    arr.add(3);  
    arr.add(10);  
    arr.add(40);  
    arr.add(2);  
    arr.add(3);  
    System.out.println(arr.size());  
    → arr.remove(2);  
    System.out.println(arr.size());  
}
```



Note :- after deleting, every element on right side will have index ↓ed by 1

Note:- adding element at a index

```
public static void main(String[] args) {  
    ArrayList<Integer> arr = new ArrayList<>();  
    arr.add(5);  
    arr.add(3);  
    arr.add(10);  
    arr.add(40);  
    arr.add(1, 7);  
}
```



# ArrayList Printing



```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    ArrayList<Integer> arr = new ArrayList<>();
    for (int i = 0; i < n; i++) {
        int val = scn.nextInt();
        arr.add(val);
    }
    printing(arr);
}
public static void printing(ArrayList<Integer> arr) {
    for (int i = 0; i < arr.size(); i++) {
        System.out.print(arr.get(i) + " ");
    }
}
```

printing of AL

→ For Each loop

Syntax

```
for (datatype var_name : ArrayList){  
    Sys0(var_name);  
}
```

Limitation:-

- for each can only traverse from left to right
- for each loop always starts at first element and always go till last element
- for each loop will only jump with 1 index
- ★ → for each loop doesn't have index

Code

```
public static void printing(ArrayList<Integer> arr) {  
    for (int i = 0; i < arr.size(); i++) {  
        System.out.print(arr.get(i) + " ");  
    }  
    System.out.println();  
  
    // for each loop  
    for (Integer ar : arr) {  
        System.out.print(ar + " ");  
    }  
}
```