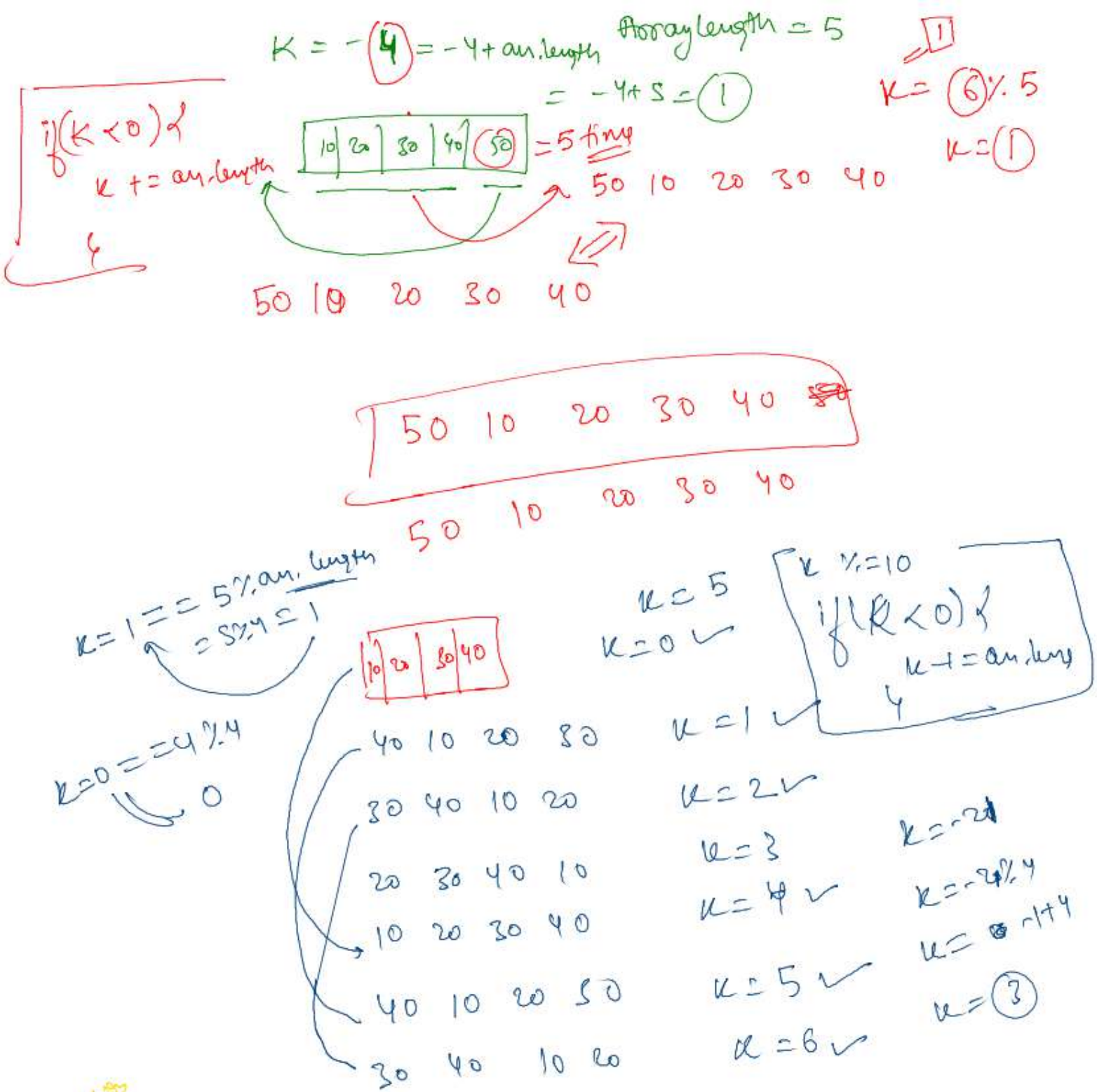

K = -(4) = -4 + arr.length
= -4 + 5 = (1)
Array length = 5
if(K < 0){
K += arr.length
}
10 20 30 40 50
= 5 times
50 10 20 30 40
50 10 20 30 40
50 10 20 30 40
K = 1 == 5 % arr.length
= 5 % 4 = 1
10 20 30 40
K = 0 == 4 % 4
0
K = 5
K = 0
K %= 10
if(K < 0){
K += arr.length
}
K = 1
40 10 20 30
K = 2
30 40 10 20
K = 3
20 30 40 10
K = 4
10 20 30 40
K = 5
40 10 20 30
K = 6
30 40 10 20
K = -24
K = -24 % 4
K = 0 + 4
K = 3

Que Triplets [i != j , j != k, k != i]

arr = [-a , -b , +c , +d , -e , -f] → target

[arr[i] + arr[j] + arr[k] == 0]

O(n^2)

[2 , -1 , -2 , 4 , 3 , 8]

[-1 , -2 , 3] = 0.

Arrays sort.

→ [-2 -1 2 3 4 8]

i left lf lr → ← right.

for(i = 0 - arr.len - 2) { left = i+1; right = arr.len - 1;

if (arr[i] + arr[left] + arr[right] == 0) { sys 0

while({

}

else if(sum < 0) {

left ++;

else right--;

if(arr[i] == arr[i-1]) {

continue;

}

```
Arrays.sort(arr);
for(int i=0;i<arr.length-2;i++){
    if(i>0 && arr[i]==arr[i-1]){
        continue;
    }
    int left =i+1;
    int right= arr.length-1;
    while(left<right){
        int sum = arr[i]+arr[left]+arr[right];
        if(sum==0){
            System.out.println(arr[i]+" "+arr[left]+" "+arr[right]);
            while(left<arr.length-1 && arr[left]==arr[left+1]){
                left++;
            }
            while(right>0 && arr[right]==arr[right-1]){
                right--;
            }
            left++;
            right--;
        }else if(sum<0){
            left++;
        }else{
            right--;
        }
    }
}
```

(arr[i] + arr[left] + arr[right] == 0)

-2 0 2 4 0 -2 +8

[-2 -2 0 0 2 4 8 7]

i k i left right

-2 -2 4

-2 0 2

arr[left] == arr[left+1]

[-2 -2 0 0 2 [4 6 7 8]]

i k k k i

left = i+1

```
if(i>0 && arr[i]==arr[i-1])
    continue;
```

4

2 0

0

sort + find

O(nlogn) + O(n²)

⇒ O(n²)

-2 -2 4

-2 0 2

Qus Quadruplet $ar[i] + ar[j] + ar[k] + ar[l] == target$

[a, b, c, d, e, f, g, h, i, j, k]

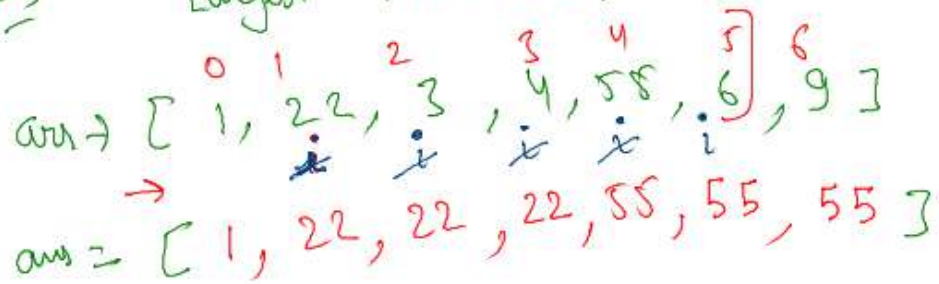
i j J [$O(n) \times n \times n$ J]

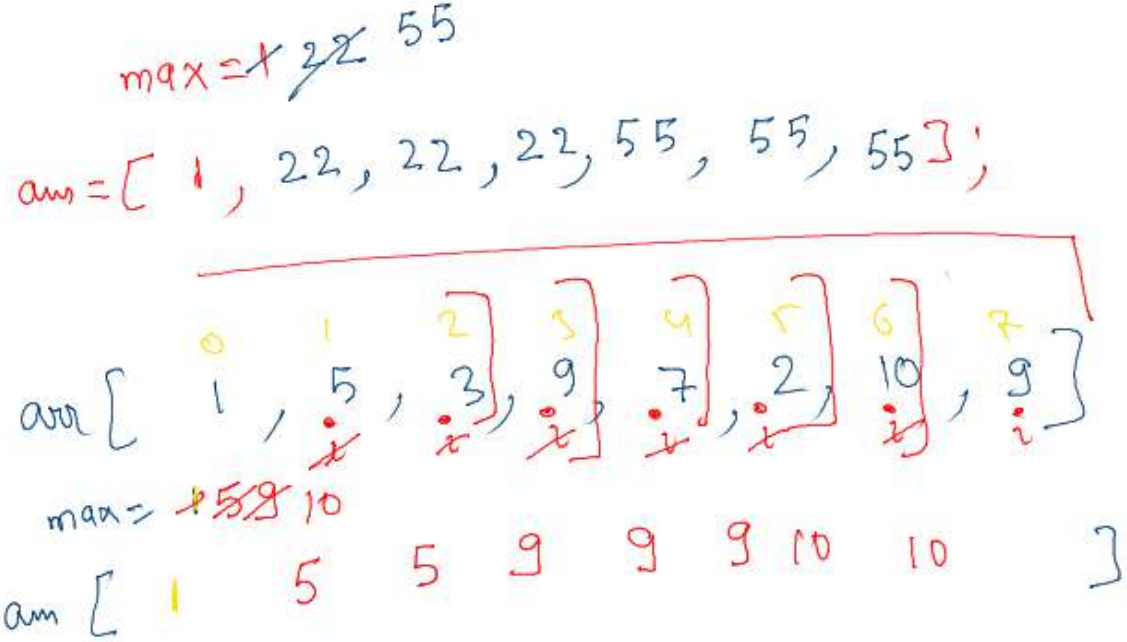
j [ ]

$O n \log n + O(n^3)$

$= O(n^3)$

Qus Largest till index.






```java
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<arr.length;i++){
        arr[i]=scn.nextInt();
    }
    int[] ans = new int[n];

    ans[0]=arr[0];
    int max = arr[0];
    for(int i=1;i<n;i++){
        max = Math.max(max,arr[i]);
        ans[i]= max;
    }
    for(int i =0;i<n;i++){
        System.out.println(ans[i]);
    }
    /* Enter your code here. Read input from STDIN. Print outp
}
```

Qu

prefix sum array

O(n)

arr = [1, 2, 3, 4, 5, 6, 7]

psarr = [1, 3, 6, 10, 15, 21, 28]

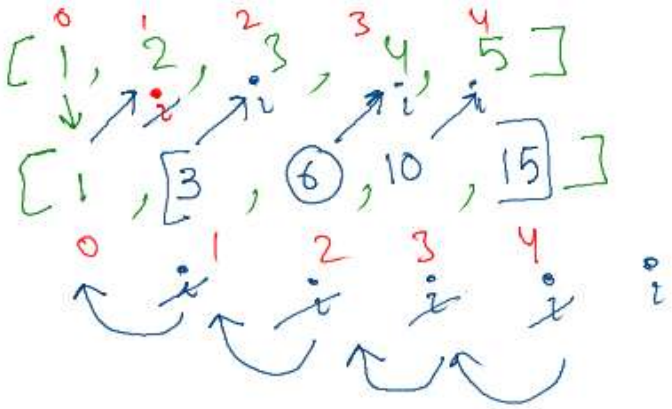
l r

[6 10 15 21]

```
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<arr.length;i++){
        arr[i]=scn.nextInt();
    }
    int left =scn.nextInt();
    int right= scn.nextInt();
    int[] ans = new int[n];
    ans[0]= arr[0];

    for(int i=1;i<arr.length;i++){
        ans[i]= ans[i-1]+arr[i];
    }


    for(int i =left;i<=right;i++){
        System.out.println(ans[i]);
    }
    /* Enter your code here. Read input from STDIN. Print out
}
```



2
6
10
15