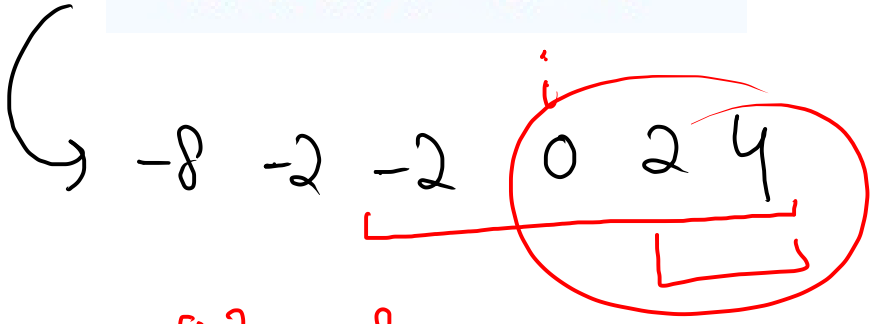
3 Sum (complete code)

```
public static void targetSum(int[] arr, int n) {
    Arrays.sort(arr);        // nlogn
    for (int i = 0; i < n; i++) {
        int target = -1 * arr[i];
        int j = i + 1;
        int k = n - 1;
        while ( j < k ) {
            int sum = arr[j] + arr[k];
            if (sum == target) {
                System.out.println(arr[i] + " " + arr[j] + " " + arr[k]);
                j++;
                k--;
            } else if (sum > target) {
                k--;
            } else {
                j++;
            }
        }
        while ( i + 1 < arr.length && arr[i] == arr[i + 1] ) i++;
    }
}
```

handle repetation for ith element

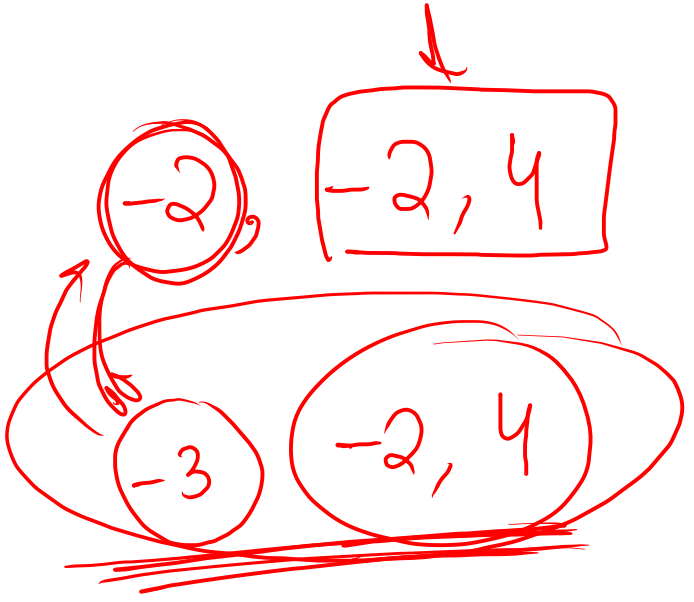
⇒ 3 Sum

```
6
-2 0 2 4 -2 -8
```



arr[i] = -8

target = 8



-2, -2, 4

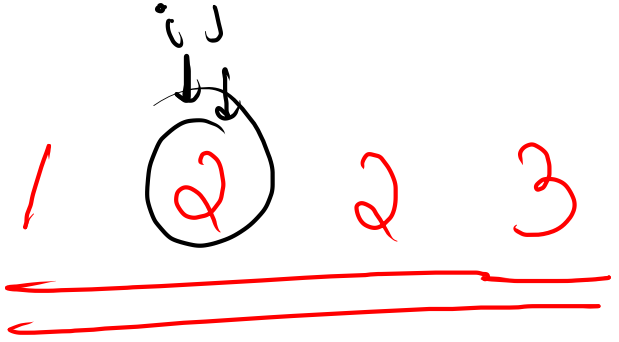
-2, 0, 2

Count boat

n = 4

arr 3 2 2 1

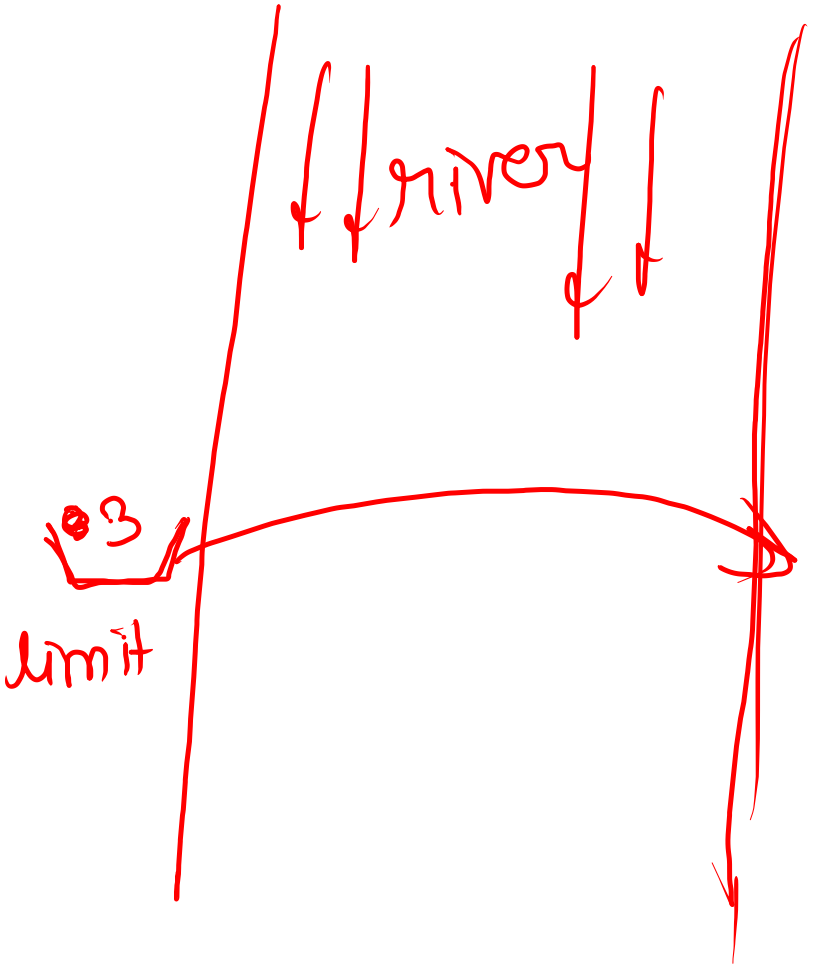
limit 3



Sum = 1+3 = 4

Sum = 1+2 = 3

i == j



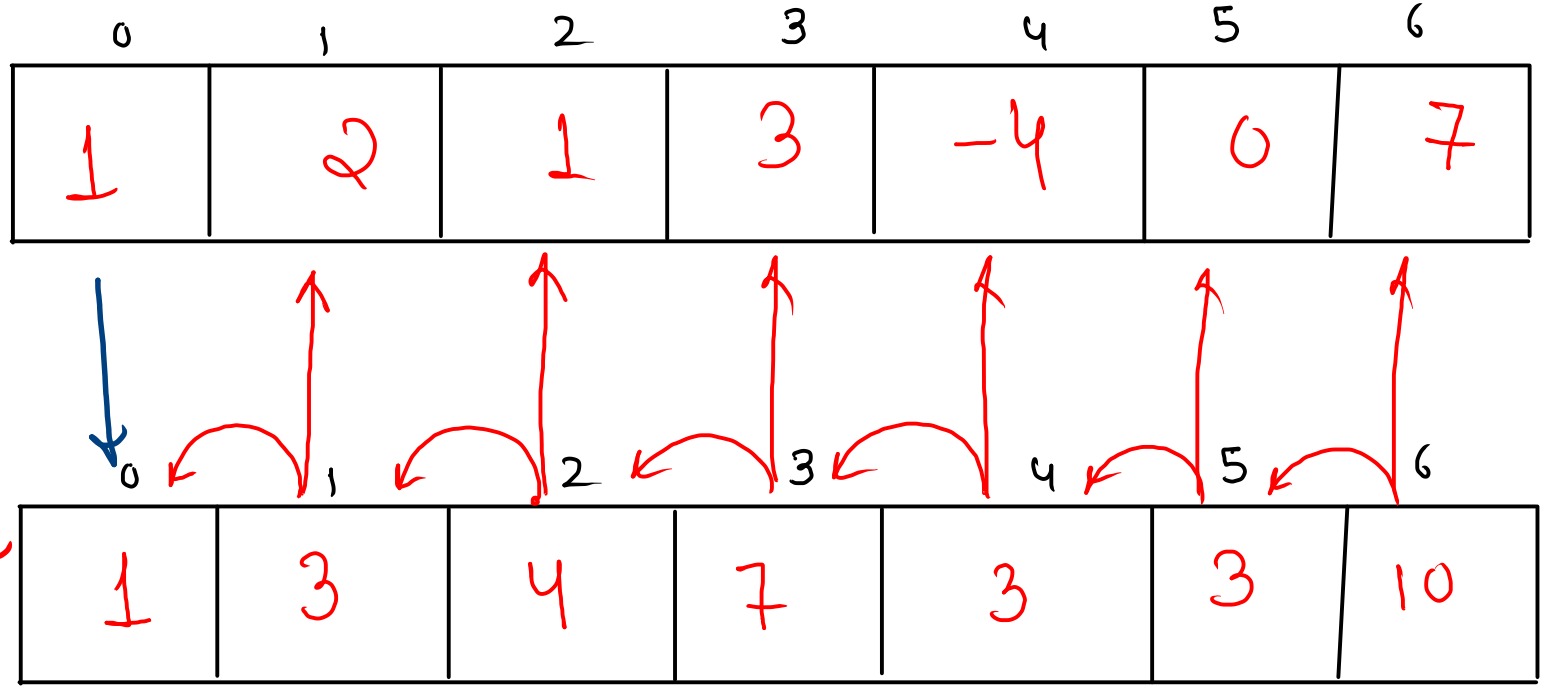
, C = ~~0~~ ~~1~~ ~~2~~ 3

Code

```java
public static int countBoat(int[] arr, int n, int limit) {
    Arrays.sort(arr);
    int i = 0;
    int j = n - 1;
    int count = 0;
    while ( i <= j ) {
        if (arr[i] + arr[j] > limit) {
            j--;    // I have crossed the j person
        } else {
            i++;    // I have crossed both persons
            j--;
        }
        count++;
    }
    return count;
}
```

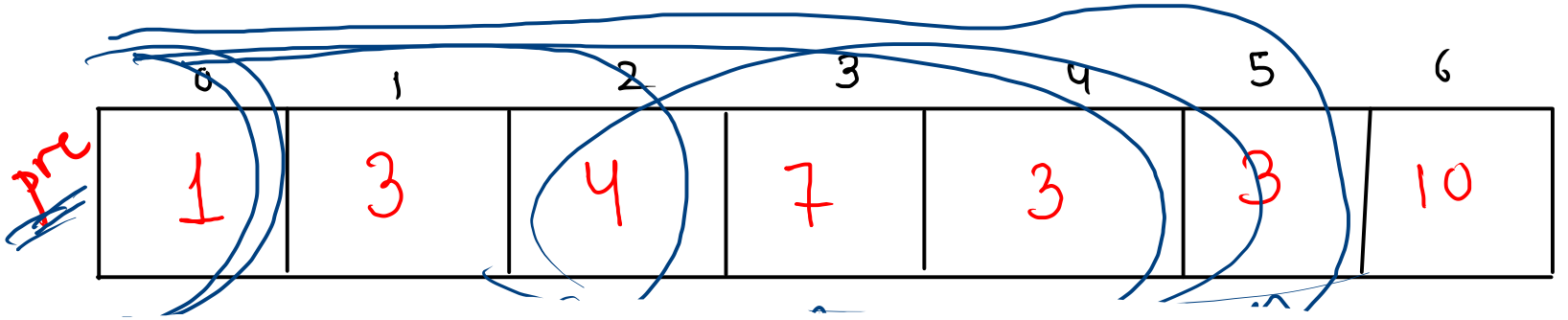
⇒ Prefix Sum

Imp



pre[i] = pre[i-1] + arr[i]

↳ where each idx contain the sum of all the previous elements



query

i = 2

j = 5

```java
public class Main {
    public static void main(String[] args) {
        int[] arr = {1, 2, 1, 3, -4, 0, 7};
        int[] prefix = new int[arr.length];
        prefix[0] = arr[0];
        for (int i = 1; i < arr.length; i++) {
            prefix[i] = prefix[i - 1] + arr[i];
        }

        int i = 3;
        int j = 5;
        int ans = prefix[j] - prefix[i - 1];
        System.out.println( ans );
    }
}
```

O(1)

i = 1

j = 4

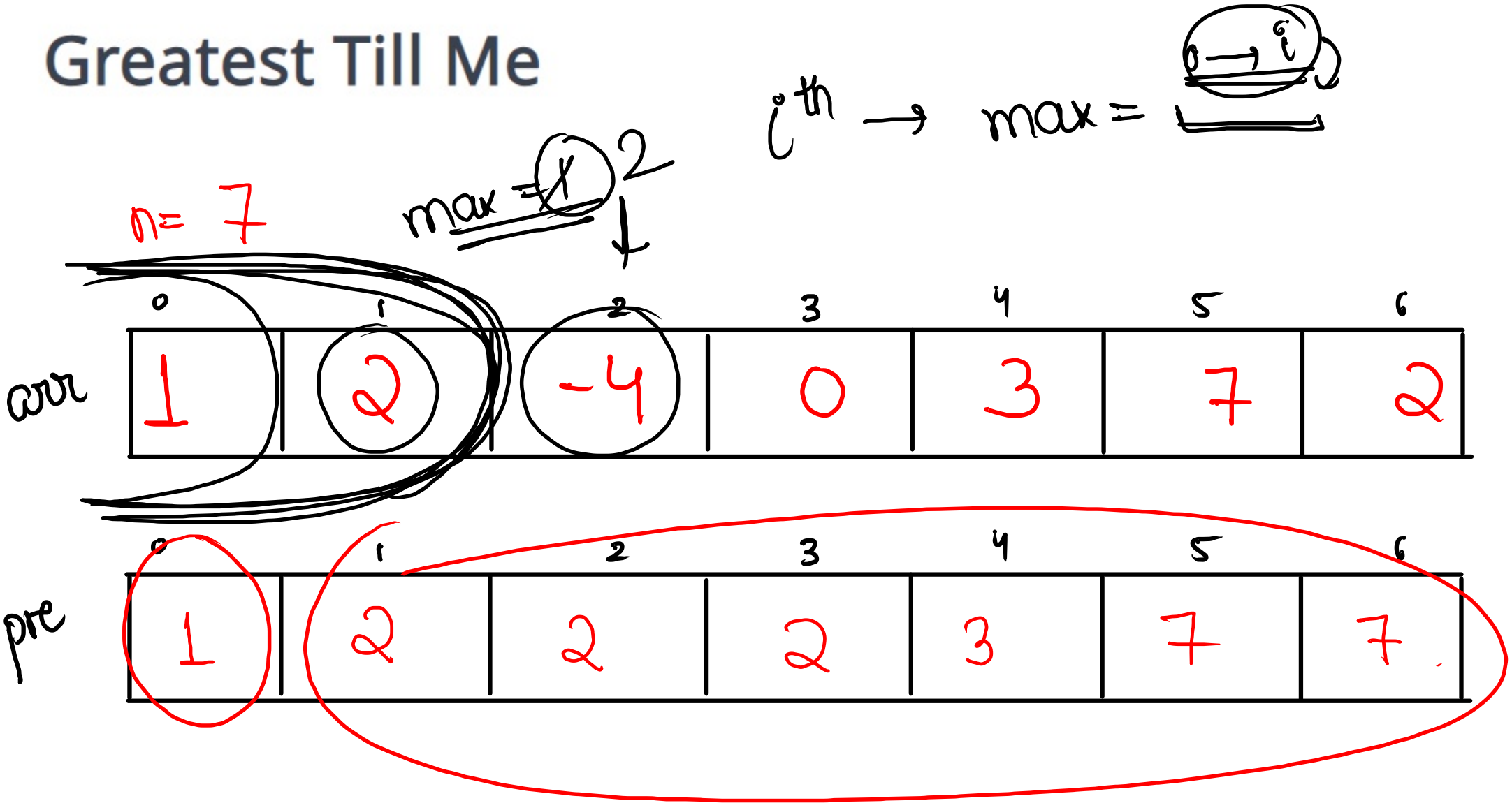
ans = pre[j] - pre[i-1]

= pre[4] - pre[0]

= 3 - 1

= 2

Greatest Till Me



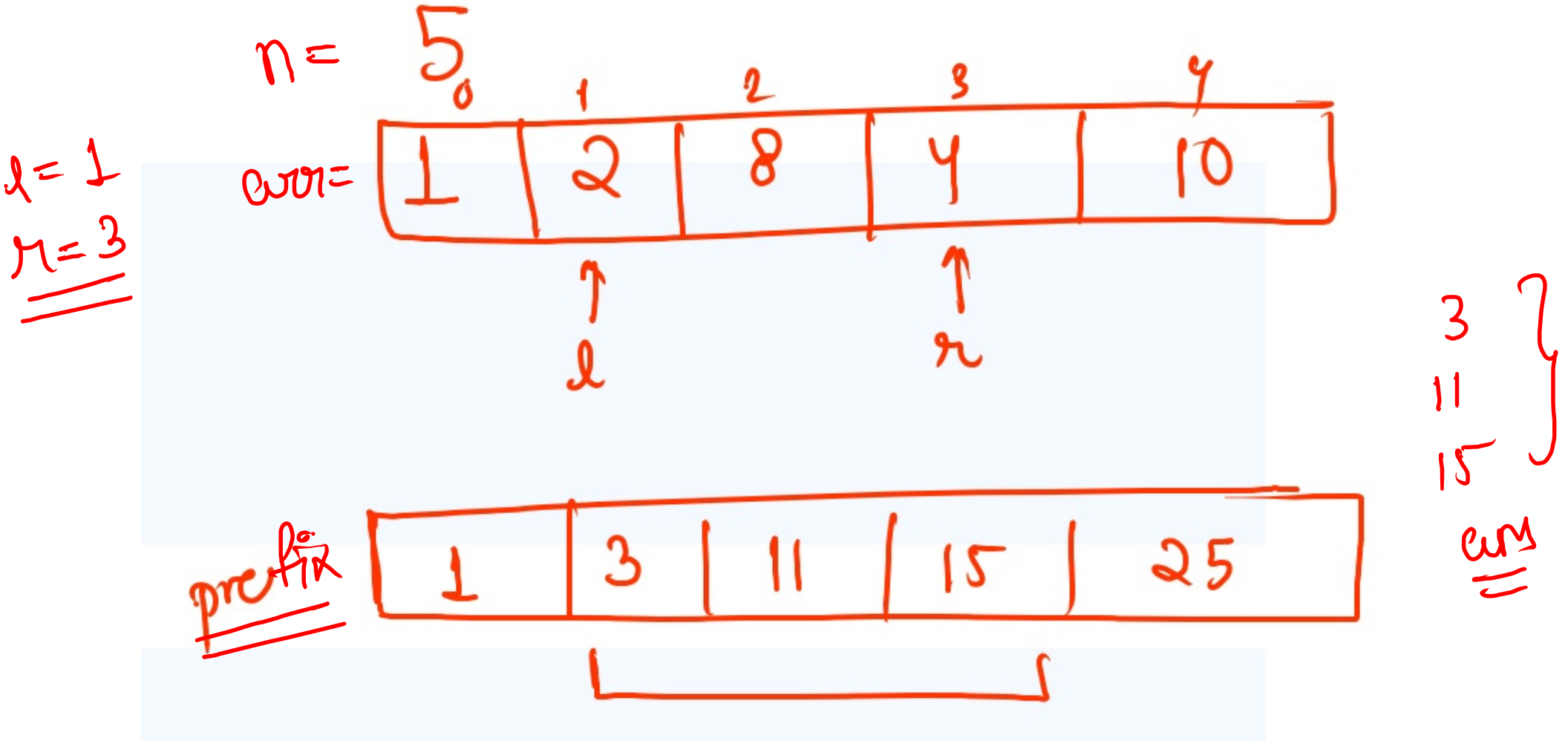
Code

```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 < n; i++) {
        arr[i] = scn.nextInt();
    }
    int[] ans = greatestTillMe(arr);
    for (int i = 0; i < n; i++) {
        System.out.println(ans[i]);
    }
}

public static int[] greatestTillMe(int[] arr) {
    int[] prefix = new int[arr.length];
    prefix[0] = arr[0];
    int max = prefix[0];
    for (int i = 1; i < arr.length; i++) {
        prefix[i] = Math.max( max, arr[i] );
        max = Math.max( max, arr[i] );
    }

    return prefix;
}
```

also possible

```java
public static int[] greatestTillMe(int[] arr) {
    int[] prefix = new int[arr.length];
    prefix[0] = arr[0];
    int max = prefix[0];
    for (int i = 1; i < arr.length; i++) {
        prefix[i] = Math.max( max, arr[i] );
        max = prefix[i];
    }

    return prefix;
}
```

Print Prefix Sum between L and R



code

```java
public static void prefixSum(int[] arr, int l, int r) {
    int[] prefix = new int[arr.length];
    prefix[0] = arr[0];
    for (int i = 1; i < arr.length; i++) {
        prefix[i] = prefix[i - 1] + arr[i];
    }

    for (int i = l; i <= r; i++) {
        System.out.println(prefix[i]);
    }
}
```