Lamda Expression

① Arrays.sort (array) → It will sort in dfault behaviour

But if you want to sort of your way then you can lamda expression

```
Integer[] arr = new Integer[n];
arr=[ 5, 3, -1, 8, 2, 6 , 4, -4]
```

① Arrays.sort (arr) → It will sort in increasing order.

② Arrays.sort (arr, (a, b)→{

return a-b / increasing
b-a / decreasing

})

a-b → increasing order

a>b a-b = +ve → a → right

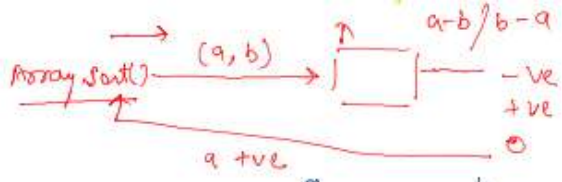
a<b a-b = -ve → a ← left

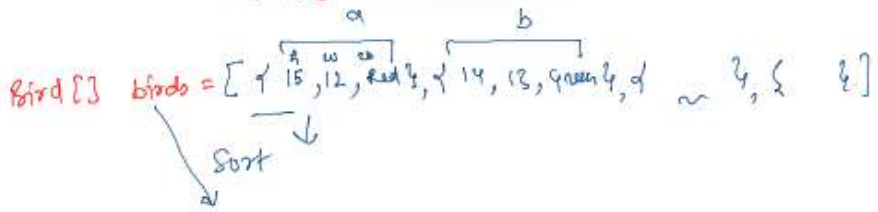
decreasing order

b>a b-a +ve b ←

b<a b-a -ve b →

a-b / b-a






```
Arrays.sort(birds, (a,b)->{
        return a.colour.length() - b.colour.length()
});
```

Ques] Reach Target

```
  0  1  2  3  4  5
[-1, 1, 2, 3, 4, 5]   target = 4
```

print all indexes whose sum of pairs is equal to target

$-1 + 5 = 4$ → [0 5]

O(n) Extra space not allowed

ans must be unique

```
  0  1  2  3  4  5  6  7  8
[10, 20, 30, 40, 50, 60, 70, 80, 90]
```

left left left left right right ~~right~~ ~~right~~ ~~right~~

right right

Sum = 90 >

target = ~~120~~ 90

Array is sorted

```
int left = 0
int right = arr.length - 1

while (left < right) {
    int sum = arr[left] + arr[right];
    if (sum == target) { sysout(left, right)
                         left++, right--; }
    else if (sum < target) { left++ }
    else if (sum > target) { right-- }
}
```

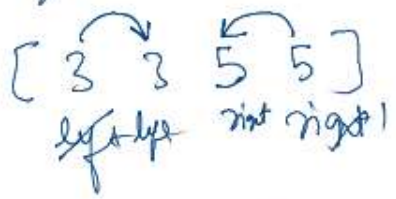
2 8
3 4

```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 target = scn.nextInt();

    int left=0;
    int right= arr.length-1;

    while(left<right){
        int sum = arr[left]+ arr[right];
        if(sum ==target){
            System.out.println(left+" "+ right);
            left++;
            right--;
        }else if(sum<target){
            left++;
        }else{
            right--;
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Y
}
```

Question [3, 3, 5, 5] target = 8

[3, 5]
↳ unique

Arrays.sort(arr);



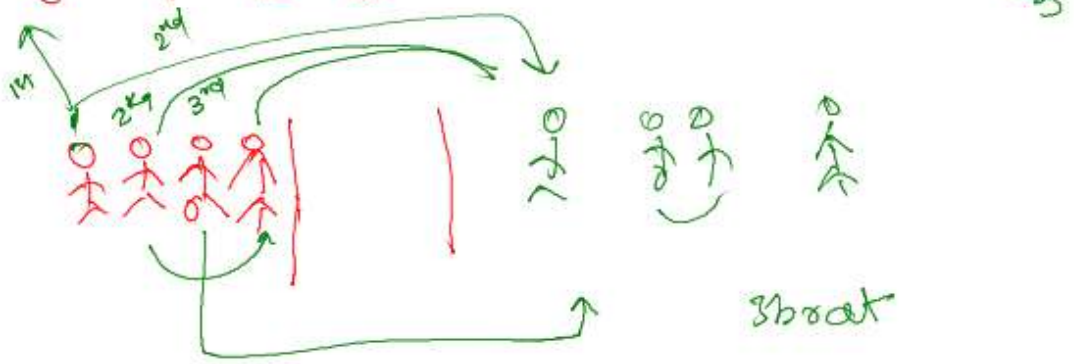
3 5

```
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 target = scn.nextInt();
    Arrays.sort(arr); // nlogn TC

    int left =0;
    int right = arr.length-1;
    while(left<right){
        int sum = arr[left]+arr[right];
        if(sum==target){
            System.out.println(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<target){
            left++;
        }else{
            right--;
        }
    }
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class shou
}
```

Qu No of boats

limit of boat = 3 kg

arr [3 , 2 , 2 , 1]
0 1 2 3



two pointer

minimum # of boats

[3 , 2 , 2 , 1]

[1 , 2 , 2 , 3]
left right right

[weight > limit]
right --;
boat ++;

O(nlogn)

boat ++;

if weight = 4 > limit.

right.

if [weight <= limit
left++;
right --;
boat ++;]

```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 limit = scn.nextInt();
    Arrays.sort(arr); // nlogn TC
    int noOfBoats=0;
    int left =0;
    int right = arr.length-1;
    while(left<=right){
        int totalweight = arr[left]+arr[right];
        if(totalweight<=limit){
            left++;
        }
        right--;
        noOfBoats++;
    }

    System.out.println(noOfBoats);
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your c
}
```