Sorting

1. Arrays.sort(arr). → it will sort in increasing order.
2. Arrays.sort(arr, Collections.reverseOrder());
   ↳ It will sort the array in decreasing order
3. Arrays.sort(arr, si, ei);
   ↳ It will sort my array in increasing Order for si (starting index) till ei (ending index)-1;

arr. →

0	1	2	3	4	5	6
10	3	7	12	6	9	4

↳ Arrays.sort(arr, 2, 6);

10	3	6	7	9	12	4
0	1	2	3	4	5	6

{ Arrays.sort(arr, 0, arr.length);
  ⇓ ↳ It will sort whole array.
  Arrays.sort(arr)

4.) Lamda Expression

4.1) Arrays.sort(arr, (a, b) → {
return a-b;
})

(a, b)

a > b → a-b (+ve)

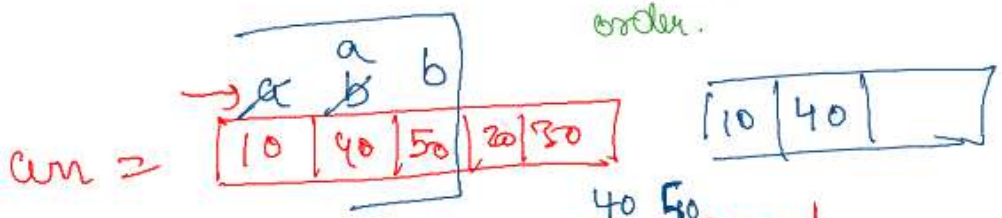
a < b → a-b (-ve)

a == b → a-b (0)

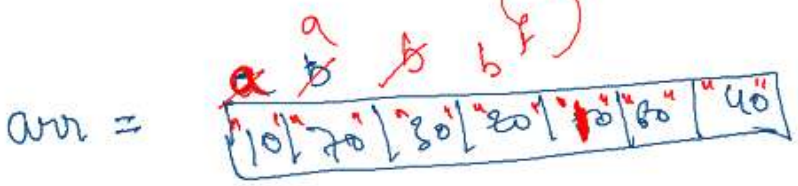
It will also sort my array in increasing order

4.2) Arrays.sort(arr, (a, b) → {
return b-a;
})

It will sort my array in decreasing order.



Arrays.sort(arr, (a, b) → {
40 50
40 - 50
return a-b;
})



Stream

a, b

a > b +ve (increasing)

a < b -ve (decreasing order)

a == b 0 (equal)

return a - b

Arrays.sort(arr, (a, b) -> {

return a - b

10 - 70 70 - 30 70 - 20

})

a ->

b ->

merge, on

Question Sort square of array

arr = [0, 6, -1, 4, -3, 2]

Squars = [0, 36, 1, 16, 9, 4]

Sort = [0, 1, 4, 9, 16, 36]

⇓

arr = [0, -1, 2, -3, 4, 6]

Arrays.sort(arr, (a, b) -> {

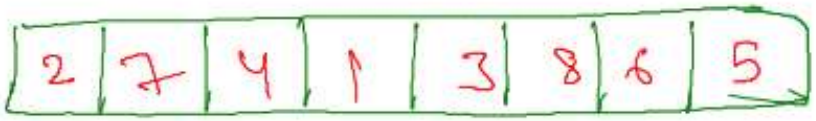
int sqa = a*a;

int sqb = b*b;

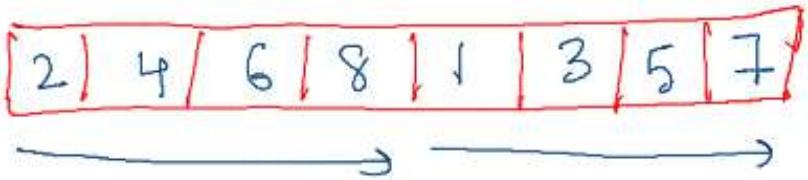
return sqa - sqb;

Ques: Sort by parity.

arr →

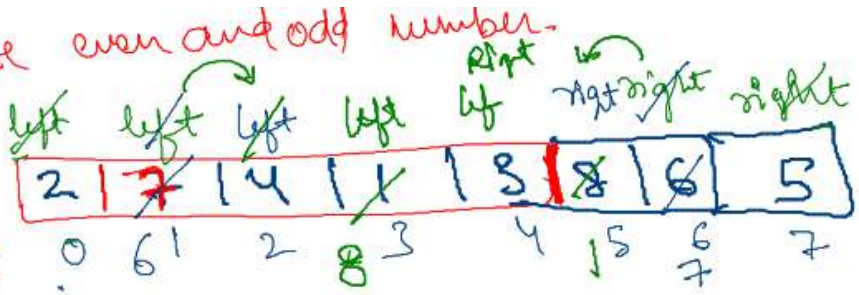


Sol



All evens in the starting and then sort in increasing order, do same for odd no followed by even highest no also in increasing order.

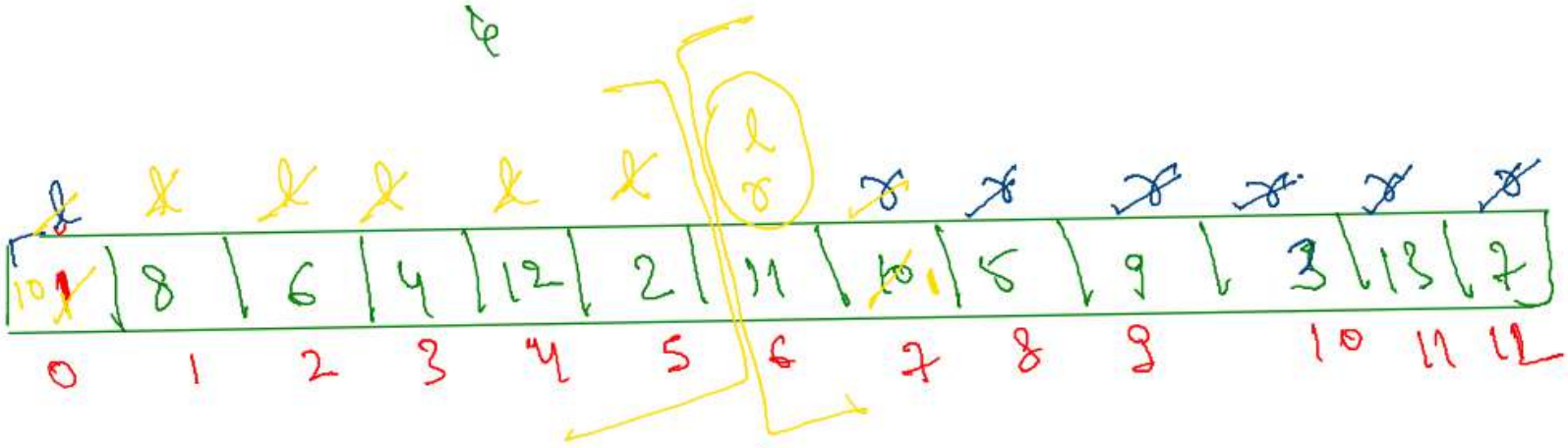
① Separate even and odd number.



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

```
if(arr[left]%2==0){
    left++;
}
```
] its already even so standing

```
else if(arr[right]%2==1){
    right--;
}
else{
    int temp = arr[left];
    arr[left] = arr[right];
    arr[right] = temp
    left++;
    right--;
}
```




```
while ( left < right) {
    if(even -> left)  left++;
    else if(right->odd)  right--;
    else {
        swap.
        left++;
        right--;
    }
}
```

```
Arrays.sort(arr, 0, left);
Arrays.sort(arr, left, arr.length);
```

```java
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 =0;
int right = arr.length-1;

while(left<right){
    if(arr[left]%2==0){
        left++;
    }else if(arr[right]%2==1){
        right--;
    }else{
        int temp = arr[left];
        arr[left]= arr[right];
        arr[right]= temp;
        right--;
        left++;
    }
}

Arrays.sort(arr,0,left);
Arrays.sort(arr,left,arr.length);

for(int i=0;i<arr.length;i++){
    System.out.print(arr[i]+" ");
}
```

O(n) + O(nlogn) + O(nlogn)

TC = O(nlogn)

O(n)

O(nlogn)

O(nlogn)