

Inbuilt Sorting

(→

↓

Arrays.sort() ;

Descending order - Using inbuilt.

Comparator.

func.

write custom logic for sorting

\* Inbuilt

[ User defined.

Summary → Comparator → sort decreasing

Java.

## Comparator

↳ can be applied only to non-primitive data type.

primitive data - type

~~int~~

~~double~~

boolean

char

short

long

byte

float

int [ ] A = new int [5];

String  
Array

Class

# Wrapper Class.

OOP

int → Integer ✓

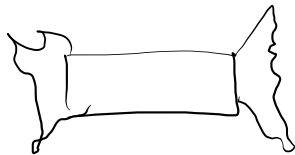
char → Character

double → Double

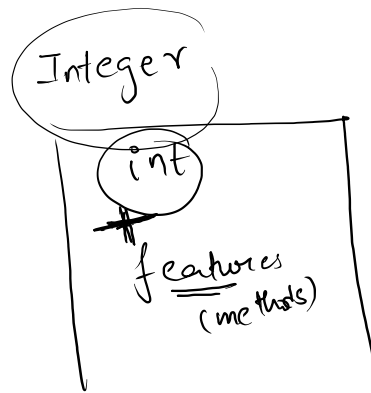
boolean → Boolean.

8

Integer.MIN\_VALUE;



velocity  
speed + direction





Auto-boxing

Un-boxing

```
1 //import java.util.*;
2 import java.util.Arrays;
3 public class Main
4 {
5     public static void main(String[] args) {
6         int [] a = new int[5];
7
8         Integer [] A = new Integer[5];
9         for(int i = 0; i < A.length; i++){
10             A[i] = scn.nextInt();
11         }
12
13
14
15     }
16 }
17
```

# decreasing order using inbuilt sort

$n = 5$       3      5      1      4 2            5 4 3 2 1

Language: Java 8

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         Integer [] A = new Integer[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        Arrays.sort(A, Collections.reverseOrder());
14        for(int i = 0; i < n; i++){
15            System.out.print(A[i] + " ");
16        }
17    }
18 }
19 }
```

Indirect.  
(Smart)

int  $\rightarrow$  1, 3, 4, 5, 2

sort  $\rightarrow$  1 2 3 4 5

reverse  $\rightarrow$  5 4 3 2 1

Inte



## Comparator.

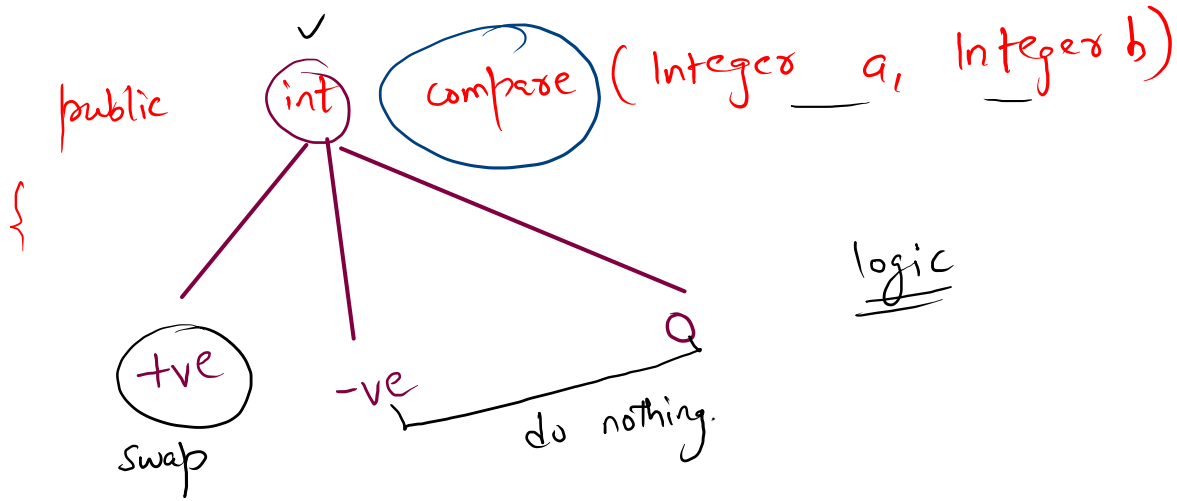


Contains a func<sup>c</sup> that holds the logic to  
compare two ~~object~~ things.

Comparator

↳ int  
compare  
# logic

```
12  
13 //how to write own comparator?  
14 Comparator<Integer> myComp = new Comparator<>(){  
15     public int compare(Integer a, Integer b){  
16         //logic  
17     }  
18 }  
19 };  
20
```



} b, a

~~a~~, ~~b~~ → +ve

a, b → -ve

\*\*\*

Explain

↳ Java T point

or

gpt

Class

Objects

Constructors

Wrapper Class

Comparator

compare  $\rightarrow$  +ve  
(swap)

descending-  
order

a b  
~~7~~ ~~2~~, ~~1~~ ~~7~~  $\rightarrow 7-2 = +ve$

9, 4  $\rightarrow b-a$   
 $4-9 = -ve$

5 5  $\rightarrow 0$

10 5  $\rightarrow 5-10 = -ve$

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         Integer [] A = new Integer[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        //how to write own comparator?
14        Comparator<Integer> myComp = new Comparator<Integer>(){
15            public int compare(Integer a, Integer b){
16                //logic
17                return b-a;
18            }
19        };
20
21        |
22        Arrays.sort(A, myComp);
23        for(int i = 0; i < n; i++){
24            System.out.print(A[i] + " ");
25        }
26
27    }
28 }
```

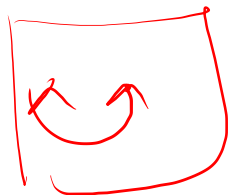
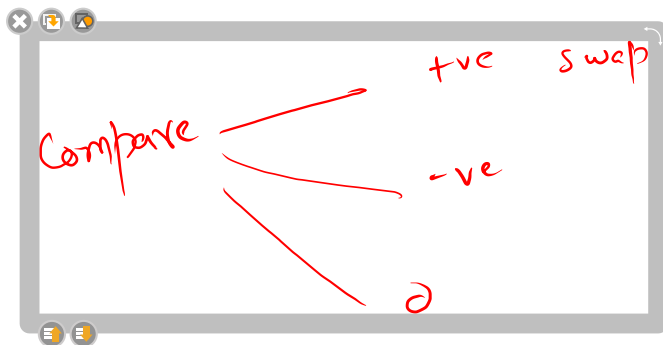
\*

$a - b \rightarrow \text{increasing}$   
 $b - a \rightarrow \text{decreasing}$

7 2

$7 - 2$

2 7



# Sort the array according to their Square of each element

Sample Input 0

```
5
4 -1 0 -5 6
```

Sample Output 0

```
0 -1 4 -5 6
```

4      -1      0      -5      6

16      1      0      25      36

increasing

→  $a - b$

$$a^2 - b^2$$

4      -5

~~-5~~      ~~4~~

4      -5

$$25 - 16 > 0$$

$$\begin{array}{|c|c|} \hline 2 & -7 \\ \hline \end{array} \quad \begin{array}{|c|c|} \hline 4 & -49 \\ \hline \end{array} = -ve$$

$\nearrow$  -ve

$$a - b$$

$\downarrow$

$$\begin{array}{cc} -7 & 2 \\ \hline 2 & -7 \\ \hline \end{array}$$

2, -7

$$2 - (-7) = +ve$$

$$\begin{array}{cc} 17 & 2 \end{array}$$

$$\begin{array}{cc} 10 & -6 \end{array}$$

$$\begin{array}{cc} 6 & 10 \end{array}$$

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         Integer [] A = new Integer[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13        //how to write own comparator?
14        Comparator<Integer> myComp = new Comparator<Integer>(){
15            public int compare(Integer a, Integer b){
16                //logic
17                return a*a - b*b;
18            }
19        };
20
21
22        Arrays.sort(A, myComp);
23        for(int i = 0; i < n; i++){
24            System.out.print(A[i] + " ");
25        }
26
27    }
28 }

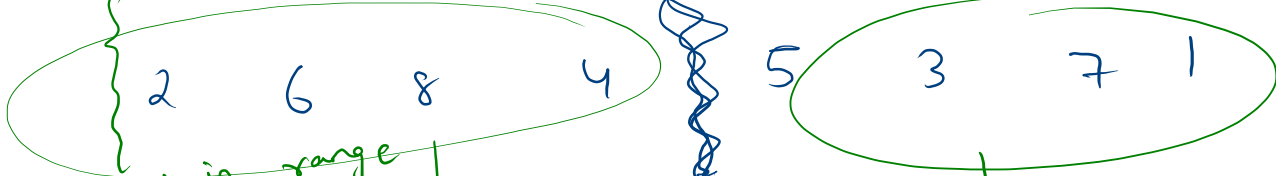
```

# Sort Array By Parity

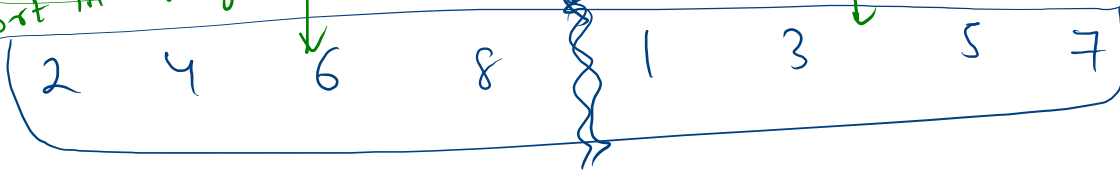
Given an integer array `nums[]`, move all the even integers at the beginning of the array followed by all the odd integers in non-decreasing order.

5      2      3      6      8      4      7      1  
         /                   /                   /                   /

1. segregation



2. { Sort in range





Sort in range.

[1, 4]

[start, end)

[1, 5)  $\rightarrow$  [1, 4]

4	3	1	7	6	8	9	5	2
0	1	2	3	4	5	6	7	8

4    1   3   6   7    8   9   5   2

```
1 // import java.util.*;
2 import java.util.Arrays;
3 public class Main
4 {
5     public static void main(String[] args) {
6         int [] A = {4,3,1,7,6,8,9,5,2};
7
8         Arrays.sort(A, 1, 5);
9
10        for(int i = 0; i < A.length; i++){
11            System.out.print(A[i] + " ");
12        }
13
14    }
15 }
16
```

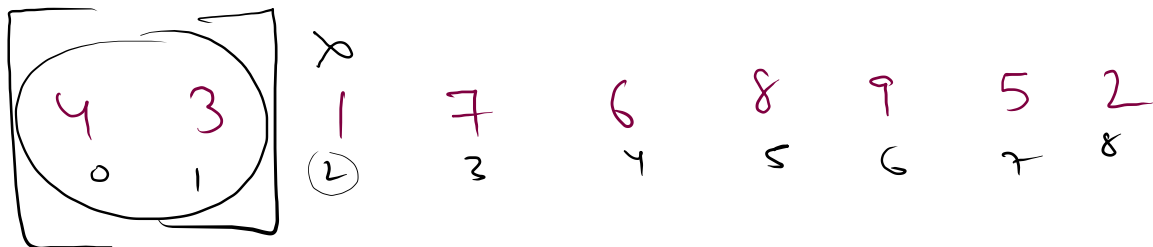
[1, 5)

[1, 4]

5

→ included

↪ excluded



res  $\rightarrow$  3 4 1 7 6 8 9 5 2  
 0 1 2 3 4 5 6 7 8

Arrays. sort(0, 2)

# Sort by Parity

Step 1: → Segregation

~~5~~<sub>0</sub> <sup>4</sup> 2  
1

~~8~~  
~~3~~  
2

6  
3

<sup>3</sup>  
~~8~~  
4  
<sub>i</sub>

<sup>5</sup>  
~~4~~  
5

7  
6

1  
2

even.... odd....

Correct position → ignore  
both wrong → swap

Pseudo code

→  $A[i] \rightarrow \text{even}$

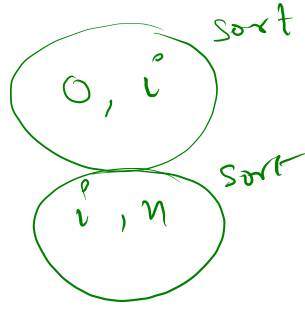
$i++$

→  $A[j] \rightarrow \text{odd}$

$j--$

→ else swap

$i++$   $j--$



```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5
6     public static void main(String[] args) {
7         Scanner scn = new Scanner(System.in);
8         int n = scn.nextInt();
9         int [] A = new int[n];
10        for(int i = 0; i < n; i++){
11            A[i] = scn.nextInt();
12        }
13
14        int i = 0;
15        int j = n-1;
16
17        while(i <= j){
18            if(A[i] % 2 == 0){
19                i++;
20            }else if(A[j] % 2 != 0){
21                j--;
22            }else{
23                int tmp = A[i];
24                A[i] = A[j];
25                A[j] = tmp;
26                i++;
27                j--;
28            }
29        }
30        Arrays.sort(A, 0, i);
31        Arrays.sort(A, i , n);
32        for(int k = 0; k < n; k++){
33            System.out.print(A[k] + " ");
34        }
35    }
36 }
```