

# → Insertion Sort

( pick the first element of unsorted array  
and insert it at the correct position )

arr = 

0	1	2	3	4
5	9	8	2	1

j = myself  
j-1 = other

arr[j] < arr[j-1]  
then swap(j, j-1)

i=0) 

5	9	8	2	1
---	---	---	---	---

i=1) 

5	9	8	2	1
5	9	8	2	1

i=2) 

5	9	8	2	1
8	9			
5	8	9	2	1

i=3) 

5	8	9	2	1
5	8	9	2	1

i=4) 

2	5	8	9	1
2	5	8	9	1

i=5) 

2	5	8	9	1
2	5	8	9	1

i=6) 

2	5	8	9	1
2	5	8	9	1

i=7) 

2	5	8	9	1
2	5	8	9	1

i=8) 

2	5	8	9	1
2	5	8	9	1

i=9) 

2	5	8	9	1
2	5	8	9	1

i=10) 

1	2	5	8	9
---	---	---	---	---

$$\text{Ex:- } \text{arr} = \begin{bmatrix} 3, -2, 6, 5, 8 \\ \dots, \dots, \dots, \dots, \dots \end{bmatrix}$$

i=0      j=0

3	-2	6	5	8
---	----	---	---	---

i=1      j=1

3	-2	6	5	8
---	----	---	---	---

i=2

-2	3	6	5	8
----	---	---	---	---

i=2

-2	3	6	5	8
----	---	---	---	---

$\uparrow j=2$

```

if (arr[j] < arr[j-1]) {
    swap(arr, j, j-1);
}
else {
    break;
}

```

i=3      j=3

-2	3	6	5	8
----	---	---	---	---

i=4

-2	3	5	6	8
----	---	---	---	---

i=4

-2	3	5	6	8
----	---	---	---	---

$\uparrow j=4$

Pseudo  
Code

```

for( int i=0 ; i < n ; i++ ) {
    for( int j=i ; j > 0 ; j-- ) {
        if (arr[j] < arr[j-1]) {
            swap(arr, j, j-1)
        }
        else {
            break;
        }
    }
}

```

# Code

```
// main logic
public static void insertionSort(int[] arr, int n) {

    for (int i = 0; i < n; i++) {
        for (int j = i; j > 0; j--) {
            if (arr[j] < arr[j - 1]) {
                swap(arr, j, j - 1);
            } else {
                break;
            }
        }
    }
}

public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

$$T.C = O(N^2)$$

where N is size  
of array

---

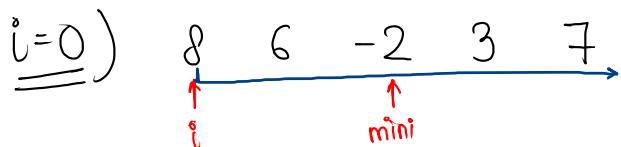
---

# $\Rightarrow$ Selection Sort

(pick the smallest element of unsorted array and place it at the start of unsorted array)

array = 

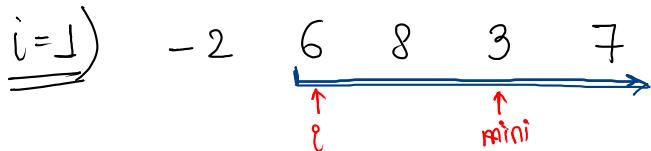
8	6	-2	3	7
0	1	2	3	4



$-2$  6 8 3 7



-2 3 6 8 7



-2 3 8 6 7



-2 3 6 7 8

psudo  
code

1)  $i^{th}$  loop traverse from 0 to  $(n-1)$

1.1)  $\text{int } \text{mini} = i;$

1.2) loop  $j^{th}$  traverse from  $i$  to  $n$

1.2-1) check if  $\text{arr}[j] < \text{arr}[\text{mini}]$   
then  $\text{mini} = j;$

1.3)  $\text{swap}(\text{arr}, \text{mini}, i);$

# Code

```
// main logic
public static void selectionSort(int[] arr, int n) {
    for (int i = 0; i < n; i++) {
        int mini = i;
        for (int j = i; j < n; j++) {
            if (arr[j] < arr[mini]) {
                mini = j;
            }
        }
        swap( arr, i, mini );
    }
}

public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}
```

$$T.C = O(N^2)$$

where N is  
size of array

$$arr = \boxed{2, -2, 3, 1, 5}$$

$i=0$ ,  $mini = \cancel{\emptyset}$ ,  $j = 0$ ,  $(2 < 2) \times$   
 $i=1$ ,  $j = 1$ ,  $(-2 < 2) \checkmark$   
 $i=2$ ,  $j = 2$ ,  $(3 < -2) \times$   
 $i=3$ ,  $j = 3$ ,  $(1 < -2) \times$   
 $i=4$ ,  $j = 4$ ,  $(5 < -2) \times$

$$\boxed{-2 \ 1} \quad 3 \quad 2 \quad 5$$

$i=2$ ,  $mini = \cancel{2}$ ,  $j = 2$ ,  $(3 < 3) \times$   
 $i=3$ ,  $j = 3$ ,  $(2 < 3) \checkmark$   
 $i=4$ ,  $j = 4$ ,  $(5 < 2) \times$

$$\boxed{-2} \quad 2 \quad 3 \quad 1 \quad 5$$

$i=1$ ,  $mini = \cancel{2}$ ,  $j = 1$ ,  $(2 < 2) \times$   
 $i=2$ ,  $j = 2$ ,  $(3 < 2) \times$   
 $i=3$ ,  $j = 3$ ,  $(1 < 2) \checkmark$   
 $i=4$ ,  $j = 4$ ,  $(5 < 1) \times$

$$\boxed{-2 \ 1 \ 2} \quad 3 \quad 5$$

$i=3$ ,  $mini = 3$ ,  $j = 3$ ,  $(3 < 3) \times$   
 $i=4$ ,  $j = 4$ ,  $(5 < 3) \times$

$$\boxed{-2 \ 1 \ 2 \ 3} \quad 5$$

Ques

All 3 sorting for ↓ing order

just change the relational  
operator from  $>$  to  $<$

## Inbuilt Sort

Arrays.sort(arr);

for ↑ing order

$$T.C = O(N \log(N))$$

Code

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

## Syntax

Arrays.sort (arr, Collection..reverseOrder());

Note :- Collections framework always work  
with wrapper classes only

Ex :-

Integer	int
Character	char
Double	double
Boolean	boolean

||

int :- 1, 2, 5, 1000, -2, -100, --

Integer :- 1, 2, 5, 1000, -2, -100, --

null

Code

$$T.C = O(n \log(n))$$

where n is size of array

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    Integer[] arr = new Integer[n];  
    for (int i = 0; i < n; i++) {  
        arr[i] = scn.nextInt();  
    }
```

```
Arrays.sort(arr, Collections.reverseOrder());
```

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

wrapper classes

Primitive datatype → Wrapper Classes

int

char

float

double

boolean

Integer

Character

Float

Double

Boolean

(String)

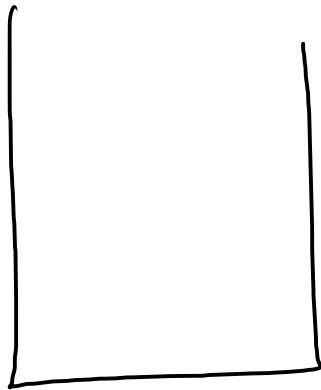
Note:-

It can store  
an extra value  
which is null

null

keyword

define



{ value = 5  
value = 0  
null

⇒ Custom sort

⇒ Lambda function ✓ gmp

Note:- definitions