

⇒ Selection Sorting (Select the smallest element from unsorted part of array and swap it with first unsorted value)

arr 

0	1	2	3	4	5
3	5	1	-2	4	0

n=6

dry run

$i=0$ ;  $j = \text{mini}$

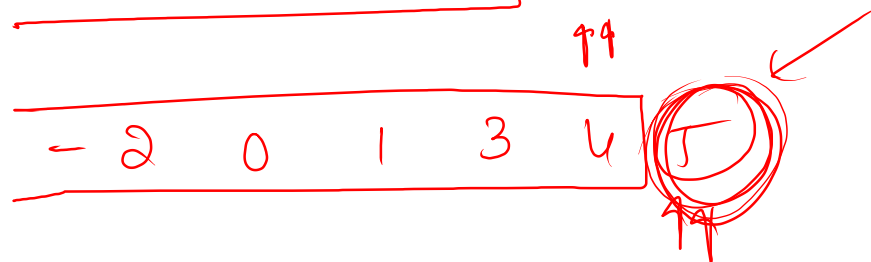
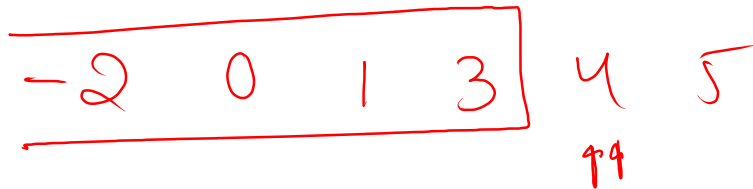
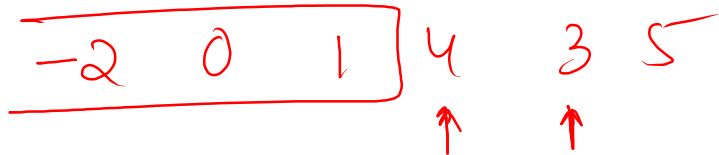
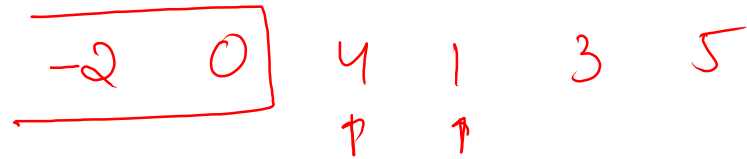
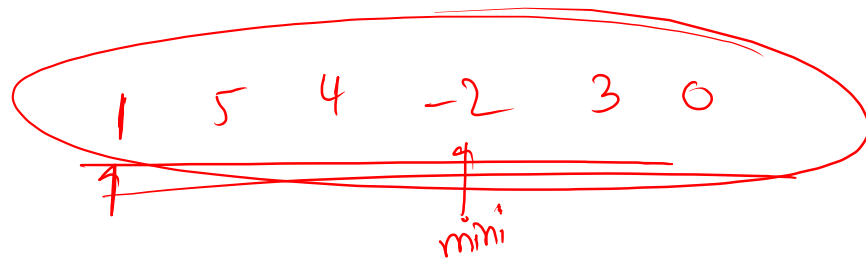
	$i \downarrow$					
	3	5	1	-2	4	0
	-2	5	1	3	4	0
		$i \downarrow$				$j \downarrow$
$i=1$ ;	-2	5	1	3	4	0
	-2	0	1	3	4	5
			$i \downarrow$			
$i=2$ ;	-2	0	1	3	4	5
	-2	0	1	3	4	5

$i=3$ ;

-2	0	1	3	4	5	
-2	0	1	3	4	5	
			$i \uparrow$	$j \uparrow$		
$i=4$ ;	-2	0	1	3	4	5
	-2	0	1	3	4	5
				$i \downarrow$	$j \downarrow$	

Code

```
public static void selectionSort(int[] arr, int n) {  
    // main logic  
    for (int i = 0; i < n - 1; i++) {  
        int mini = i;    // index  
        for (int j = i + 1; j < n; j++) {  
            if (arr[mini] > arr[j]) {  
                mini = j;  
            }  
        }  
        swap(arr, mini, i);  
    }  
  
    // print array  
    for (int i = 0; i < n; i++) {  
        System.out.print(arr[i] + " ");  
    }  
}
```



Note:- In selection sort, last element is always sorted.

# HW\_Kth Smallest Element

```
public static int kthSmallest(int[] arr, int n, int k) {  
    // sorting the array  
    bubbleSort(arr, n);  
    // print (k-1)th element  
    return arr[k - 1];  
}
```

```
public static void bubbleSort(int[] arr, int n) {  
    for (int i = 1; i < n; i++) {  
        for (int j = 0; j < n - i; j++) {  
            if (arr[j] > arr[j + 1]) {  
                swap(arr, j, j + 1);  
            }  
        }  
    }  
}
```

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

if bubble sort is in  
ascending order

and

if we sort in descending  
order then return

arr[n - K];

⇒ Inbuilt function for sorting the  
array

→ Arrays.sort(arr); // only for  
ascending order

```
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] + " ");  
    }  
}
```

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

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

↳ Comparator & Comparable

↳ lambda function