⇒ Selection Sort (pick the smallest element of unsorted array and place it in start of unsorted array)

arr =

8	6	-2	3	7
0	1	2	3	4

Dry run

1) i=0

i ↓ , mini ↓

~~8~~ 6 ~~-2~~ 3 7

-2 (8)

-2 6 8 3 7

2) i=1

i ↓ , mini ↓

-2 ~~6~~ 8 ~~3~~ 7

(3) (6)

-2 3 8 6 7

3) i=2 (2)

i ↓ , mini ↓

-2 3 ~~8~~ ~~6~~ 7

(6) (8)

-2 3 6 8 7

4) i=3

i ↓ , mini ↓

-2 3 6 ~~8~~ ~~7~~ 8

(7) (8)

-2 3 6 7 8

psudo code

```
for (int i=0 ; i < n-1 ; i++){ //(n-1) times
    int mini = i ;
    for ( int j= i ; j < n ; j++) {
        if( arr[j] < arr[mini] ){
            mini = j ;
        }
    }
    swap( arr , i , mini) ;
}
```

code

```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();
    }

    selectionSort(arr, n);
}
// main logic
public static void selectionSort(int[] arr, int n) {

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

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

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

future

- merge sort
- quick sort

count sort

radix sort

heap sort

⇒ for decreasing order

previously checking for larger element

now, check for smaller element

⇒ Inbuilt Sort function

T.C

// O(N log(N))

1) Arrays.sort(arr_name);

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

    // in built function
    Arrays.sort(arr);

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

[only for increasing order]

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

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

    // in built function
    Arrays.sort(arr, Collections.reverseOrder());

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

[for decreasing order]