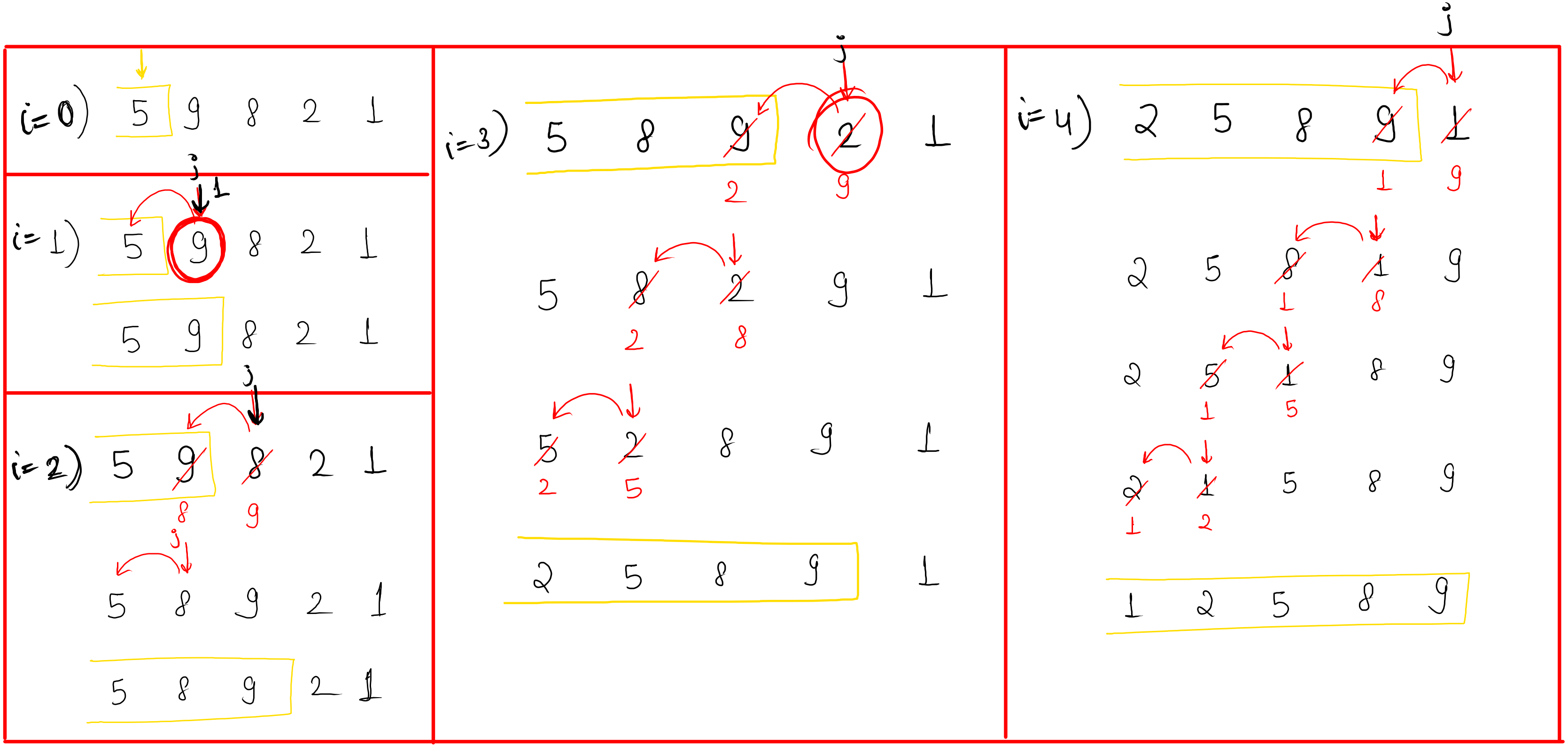
⇒ Insertion sort

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

arr =

0	1	2	3	4
5	9	8	2	1

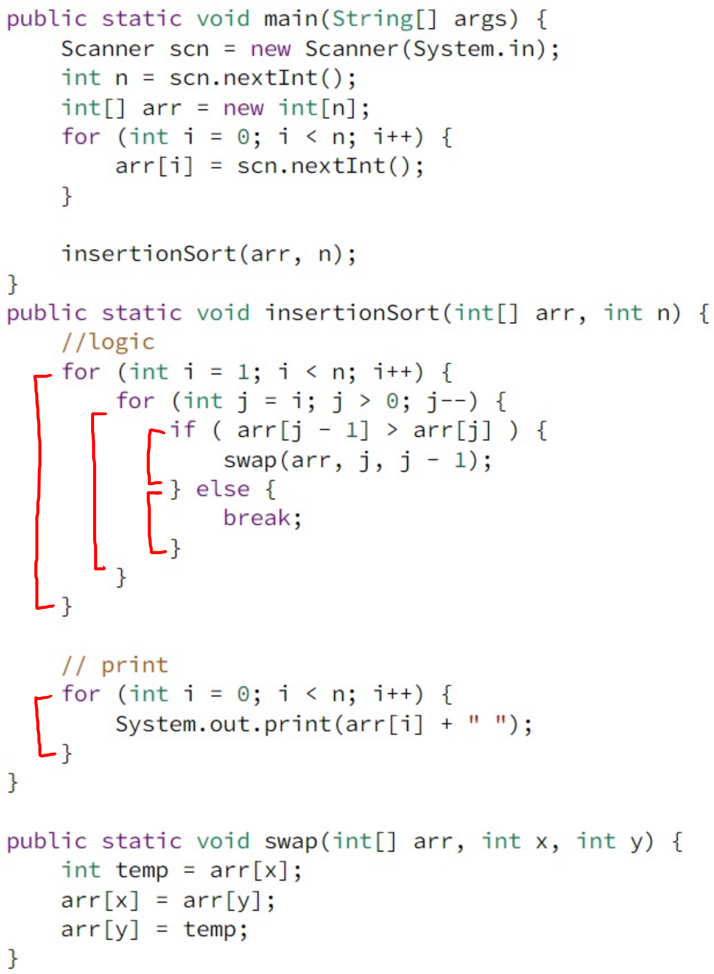
, n = 5



psudo code

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

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();
    }

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

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

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

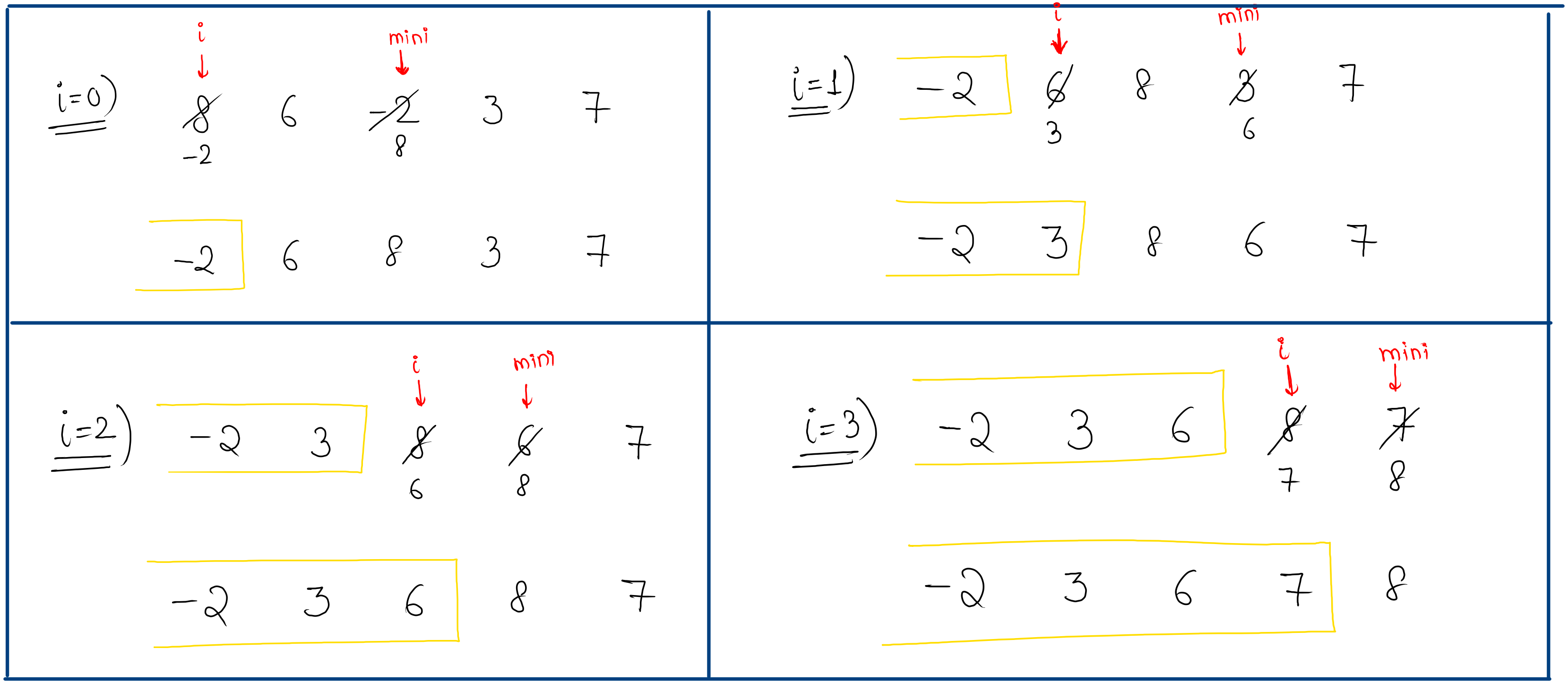
⇒ Selection Sort

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

arr =

0	1	2	3	4
8	6	-2	3	7

(i will always be at the start of unsorted array)

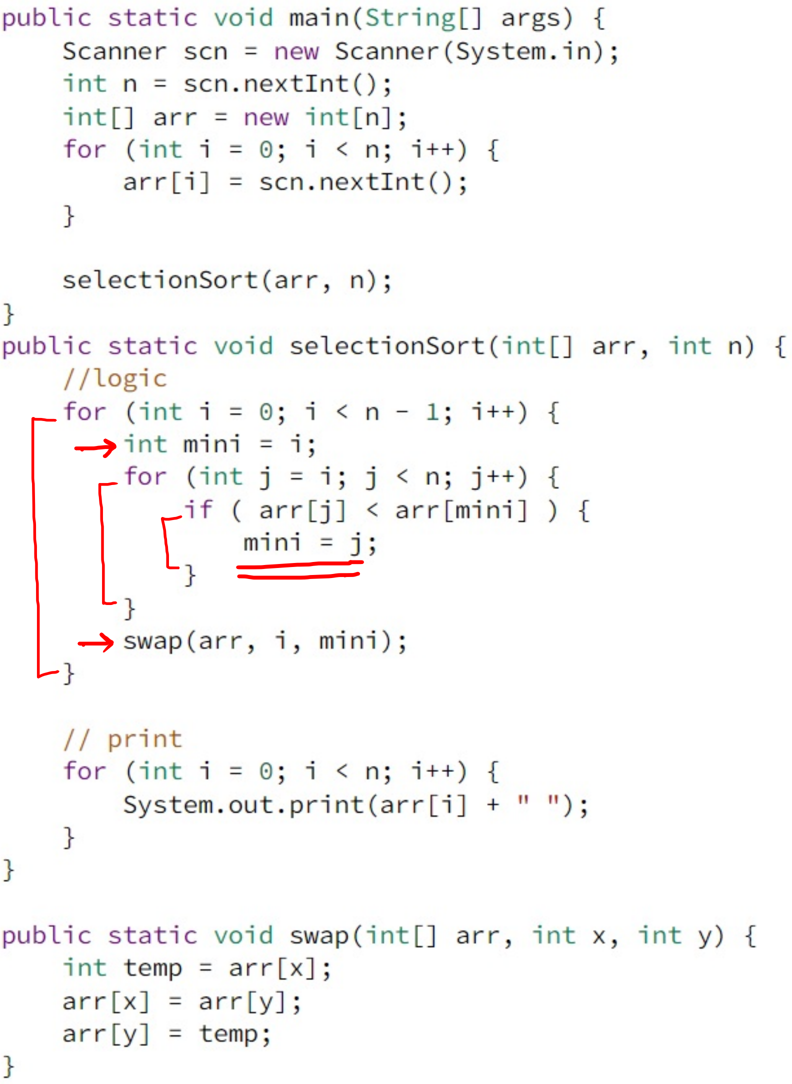


psudo code)

```
1) traverse from 0 to n-1
      int mini = -1;
   1.1) traverse from i to n
        1.1.1) if ( arr[j] < arr[mini] )
                    mini = j

   1.2) swap ( i , mini )
```

Code

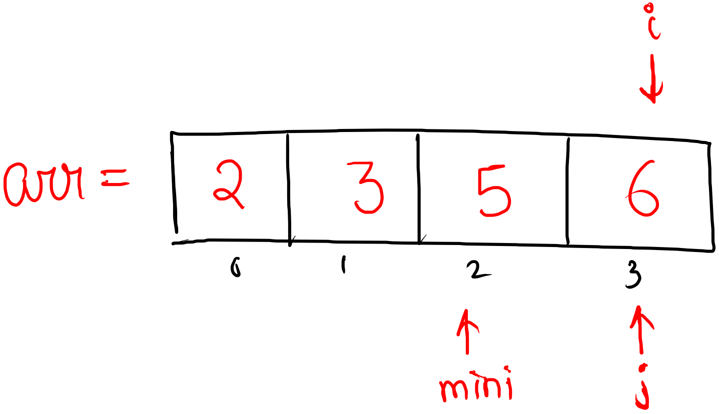


```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

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

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

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



i = 0, j = 0 , (5 < 5)

j = 1, (3 < 5) ✓

j = 2, (2 < 3) ✓

j = 3, (6 < 2)

---

i = 1, j = 1 (3 < 3)

j = 2 (5 < 3)

j = 3 (6 < 3)

---

i = 2, j = 2 (5 < 5)

j = 3 (6 < 5)

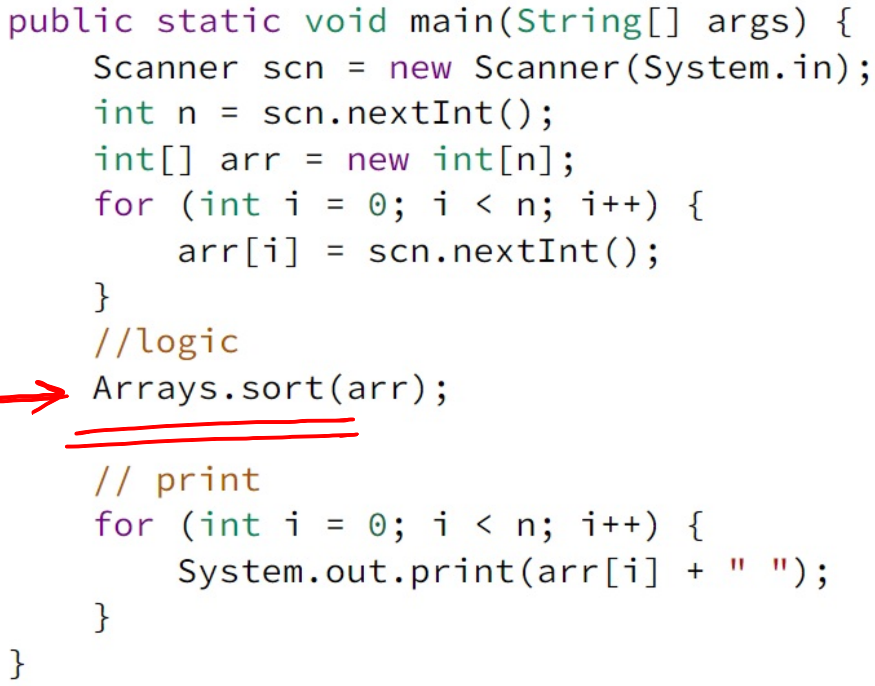
$\Rightarrow$ for decreasing order

just change sign under if condⁿ for all 3 algorithms

⇒ Inbuilt function

int[] arr = {3, 2, 7, -4, 0}

Arrays.sort(arr);



```
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();
    }
    //logic
    Arrays.sort(arr);

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

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

where, n is size of array

H.W

find out, how to change default functionality of inbuilt sort to decreasing order.