Sorting

07 August 2023 13:17

Selection Sort → Find min → place it at the correct position.

$n = 5$ → again, $n-1$ iterations

[7, 4, 2, 3, -2]

i → Place for which suitable candidate (min) is being found

j → Traverse the array and try to find the minimum

m → Current best minimum so far

1.

1.1 $\underset{\cancel{m}}{\overset{i}{7}}\ \ \underset{m}{\overset{j}{4}}\ \ 2\ \ 3\ \ -2$

ques → does j denotes a better minimum than m?

Yes

↳ so, m will change it's position and come to j

1.2 $\overset{i}{7}\ \ \underset{\cancel{m}}{4}\ \ \underset{m}{\overset{j}{2}}\ \ 3\ \ -2$

1.3 $\overset{i}{7}\ \ 4\ \ \underset{m}{2}\ \ \overset{j}{3}\ \ -2$

1.4 $\overset{i}{7}\ \ 4\ \ \underset{\cancel{m}}{2}\ \ 3\ \ \underset{m}{\overset{j}{-2}}$

Settlement $\boxed{-2}\ \ 4\ \ 2\ \ 3\ \ 7$ one element is now sorted.

Swap elements at i and m

2.)

2.1) $\boxed{-2}\ \ \underset{\cancel{m}}{\overset{i}{4}}\ \ \underset{m}{\overset{j}{2}}\ \ 3\ \ 7$

2.2) $\boxed{-2}\ \ \overset{i}{4}\ \ \underset{m}{2}\ \ \overset{j}{3}\ \ 7$

2.2) $\boxed{-2}\ \ 4\ \ \underset{m}{2}\ \ 3\ \ 7$

2.3) $\boxed{-2}\ \ \overset{i}{4}\ \ \underset{m}{2}\ \ 3\ \ \overset{j}{7}$

$\boxed{-2\ \ 2}\ \ 3\ \ 4\ \ 7$

3.

3.1) $\boxed{-2\ \ 2}\ \ \overset{i}{\underset{m}{3}}\ \ \overset{j}{4}\ \ 7$

3.2) $\boxed{-2\ \ 2}\ \ \overset{i}{\underset{m}{3}}\ \ 4\ \ \overset{j}{7}$

$\boxed{-2\ \ 2\ \ 3}\ \ 4\ \ 7$

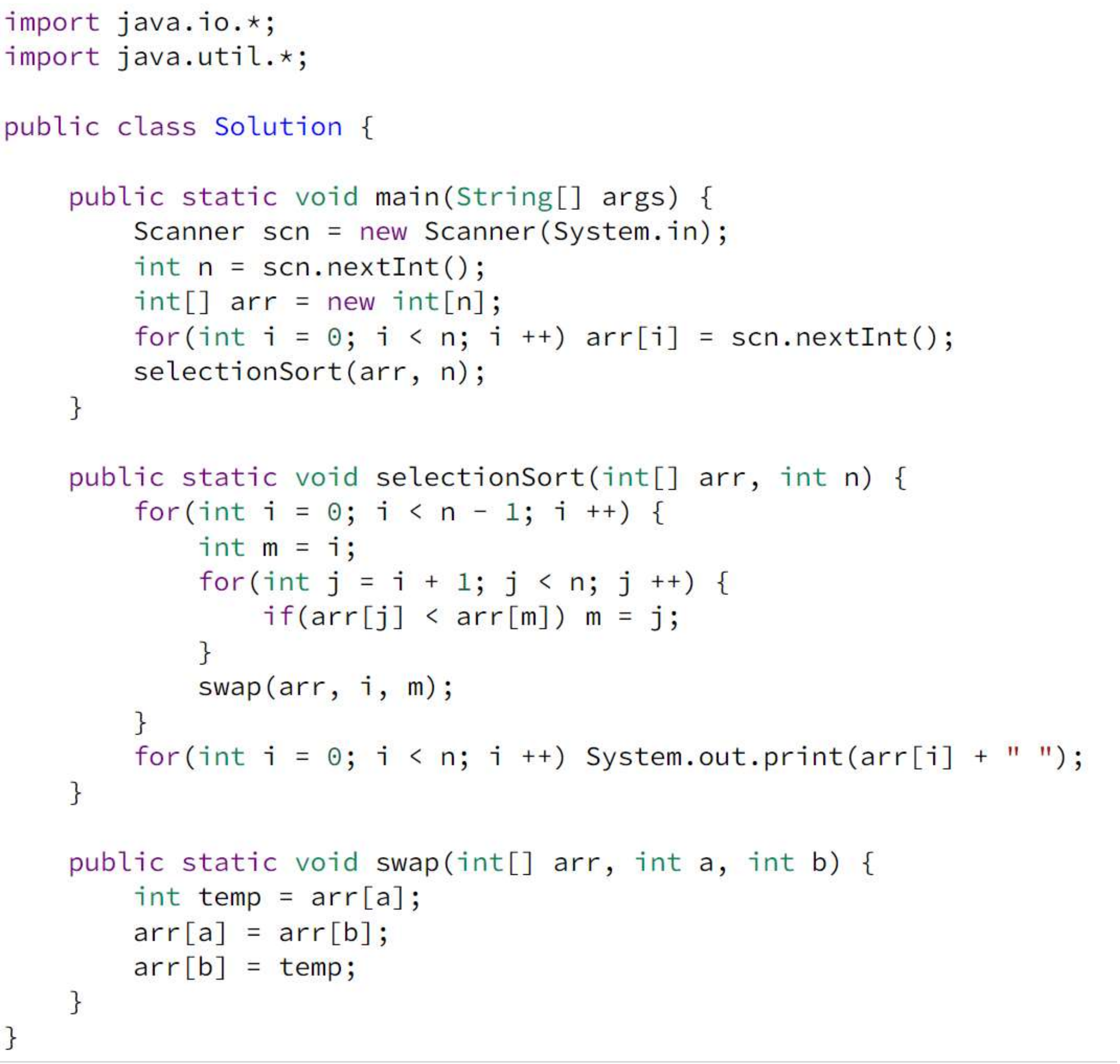
4)

4.1) $\boxed{-2\ \ 2\ \ 3}\ \ \overset{i}{\underset{m}{4}}\ \ \overset{j}{7}$

$\boxed{-2\ \ 2\ \ 3\ \ 4}\ \ 7$

* [−2 2 3 4 7] → final array.

Code



```java
import java.io.*;
import java.util.*;

public class Solution {

    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) {
        for(int i = 0; i < n - 1; i ++) {
            int m = i;
            for(int j = i + 1; j < n; j ++) {
                if(arr[j] < arr[m]) m = j;
            }
            swap(arr, i, m);
        }
        for(int i = 0; i < n; i ++) System.out.print(arr[i] + " ");
    }

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

<u>Insertion Sort</u> → Pick a number → try to insert it into the sorted array.

1. 1.1) $\boxed{7}$ 4 2 3 −2 → here, we will assume that first no denotes a sorted array and will try to insert all other nos in.

$\boxed{4\ 7}$ 2 3 −2

2. 2.1 $\boxed{4\ 7}$ 2 3 −2

2.2 4 2 7 3 -2

$\boxed{2\ 4\ 7}$ 3 -2

3.

3.1) $\boxed{2\ 4\ 7}$ 3 -2

3.2) 2 4 3 7 -2

3.3) 2 3 4 7 -2

$\boxed{2\ 3\ 4\ 7}$ -2

4.

4.1 $\boxed{2\ 3\ 4\ 7}$ -2

4.2 2 3 4 -2 7

4.3 2 3 -2 4 7

4.4 2 -2 3 4 7

-2 2 3 4 7