15-Dec-2021

15 December 2021 19:34

Sorting :-

[3, 1, 5, 2, 4] => [1, 2, 3, 4, 5]

Bubble Sort :-

0 1 2 3 4 5

5, 4, 3, 2, 1, 0

4, 5, 3 2 1 0

4, 3, 5, 2, 1, 0

4, 3, 2, 5, 1, 0

4, 3, 2, 1, 5, 0

4, 3, 2, 1, 0, 5

i = 5 | i-1 = 4

j = arr.length - 1 to 1

for(i=1 to j) {

if (arr[i] < arr[i-1]) {

swap(i, i-1)

}

j = 0

i =

0 1 2 3 4 5

[5, 4, 3, 2, 1, 0]

4 5 3 2 1 0

4 3 5 2 1 0

4 3 2 5 1 0

4 3 2 1 5 0

4 3 2 1 0 $\boxed{5}$

3 4 2 1 0 5

3 2 4 1 0 5

3 2 1 4 0 5

3 2 1 0 $\boxed{4\ 5}$

2 3 1 0 4 5

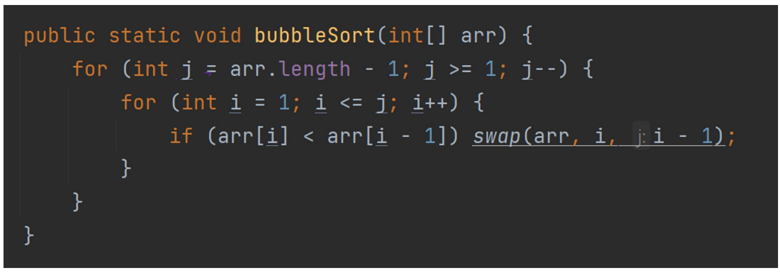
2 1 3 0 4 5

2 1 0 $\boxed{3\ 4\ 5}$

1 2 0 3 4 5

1 0 $\boxed{2\ 3\ 4\ 5}$

0 1 2 3 4 5



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

[5, 4, 3, 2, 1, 0]

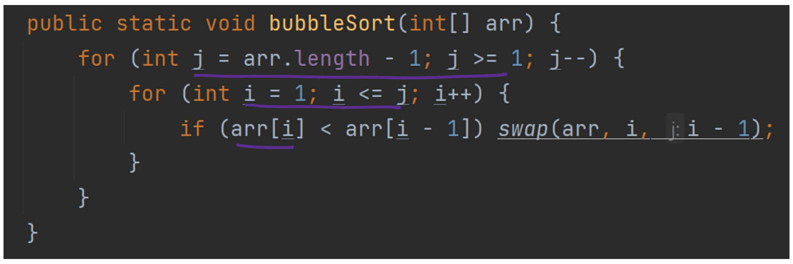
0 1 2 3 4 5

0 1 2 3

2 1 4 3

1 2 4 3

$\boxed{1\ 2\ 3\ 4}$



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

$O(n^2)$

Selection Sorting :-

3 , 2 , 1 , 0

i

$\boxed{0}$ 2 1 3

i

0 1 2 3

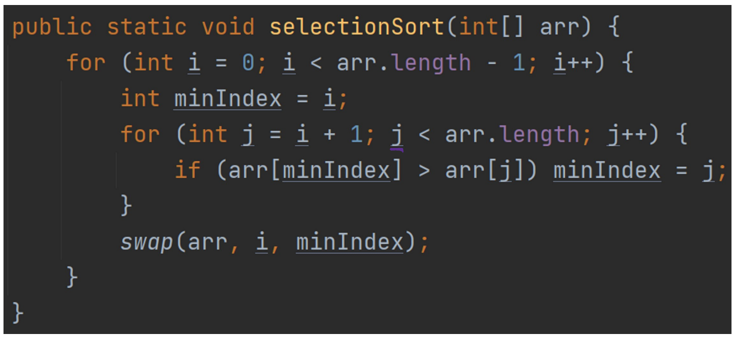
4 , 3 , 2 , 1 , 0

0 1 2 3 4

$\boxed{0}$, 3 , 2 , 1 , 4

i M

0 1 2 3 4



```java
public static void selectionSort(int[] arr) {
    for (int i = 0; i < arr.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < arr.length; j++) {
            if (arr[minIndex] > arr[j]) minIndex = j;
        }
        swap(arr, i, minIndex);
    }
}
```

j = 4 minIdx = 3

j =

Insertion Sorting :-

0 1 2 3 4

5 | 4 3 2 1

4 5 | 3 2 1

4 3 5 2 1

3 4 5 | 2 1

3 4 2 5 1

3 2 4 5 1

2 3 4 5 | 1

2 3 4 1 5

2 3 1 4 5

```
for(i=1 to arr.length-1) {
    for(j=i; j ≥ 1; j--) {
        if (arr[j] < arr[j-1])
            swap(arr, j, j-1)
} else break;
    i=   , j= 0
```

2 3 4 1 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

2, 3, 4, 5, 6, 7, 8, 9,

0 1 2 3 4 5 6 7 8

2 3 4 5 6 7 8 1 9

2 3 4 5 6 7 1 8 9

2 3 4 5 6 1 7 8 9

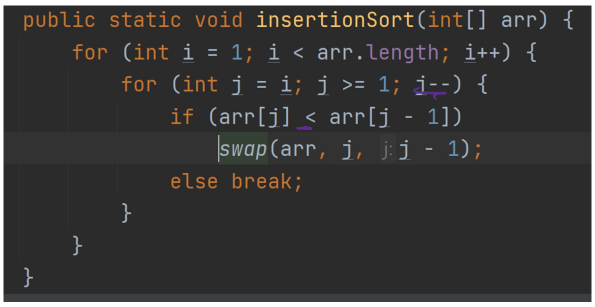
2 3 4 5 1 6 7 8 9

2 3 4 1 5 6 7 8 9

2 3 1 4 5 6 7 8 9

2 1 3

1 2 3 4 5 6 7 8 9



```java
public static void insertionSort(int[] arr) {
    for (int i = 1; i < arr.length; i++) {
        for (int j = i; j >= 1; j--) {
            if (arr[j] < arr[j - 1])
                swap(arr, j, j - 1);
            else break;
        }
    }
}
```

$O(n^2)$

Worst = (n^2)

Best = $O(n)$