Insertion, Selection or Bubble?

Below is given the original array and an entire run of a sorting algorithm (always at the top of the outer loop). Identify the algorithm.

```
[46, 74, 38, 37, 39, 55, 14, 76, 22, 62]
[46, 74, 38, 37, 39, 55, 14, 76, 22, 62]
[46, 74, 38, 37, 39, 55, 14, 76, 22, 62]
[38, 46, 74, 37, 39, 55, 14, 76, 22, 62]
[37, 38, 46, 74, 39, 55, 14, 76, 22, 62]
[37, 38, 39, 46, 74, 55, 14, 76, 22, 62]
[37, 38, 39, 46, 55, 74, 14, 76, 22, 62]
[14, 37, 38, 39, 46, 55, 74, 76, 22, 62]
[14, 37, 38, 39, 46, 55, 74, 76, 22, 62]
[14, 22, 37, 38, 39, 46, 55, 74, 76, 62]
[14, 22, 37, 38, 39, 46, 55, 62, 74, 76]

[22, 45, 0, 10, 54, 46, 55, 36, 15, 94]
[0, 45, 22, 10, 54, 46, 55, 36, 15, 94]
[0, 10, 22, 45, 54, 46, 55, 36, 15, 94]
[0, 10, 15, 45, 54, 46, 55, 36, 22, 94]
[0, 10, 15, 22, 54, 46, 55, 36, 45, 94]
[0, 10, 15, 22, 36, 46, 55, 54, 45, 94]
[0, 10, 15, 22, 36, 45, 55, 54, 46, 94]
[0, 10, 15, 22, 36, 45, 46, 54, 55, 94]
[0, 10, 15, 22, 36, 45, 46, 54, 55, 94]
[0, 10, 15, 22, 36, 45, 46, 54, 55, 94]
[0, 10, 15, 22, 36, 45, 46, 54, 55, 94]

[4, 96, 84, 75, 11, 94, 93, 57, 36, 19]
[4, 96, 84, 75, 11, 94, 93, 57, 36, 19]
[4, 11, 84, 75, 96, 94, 93, 57, 36, 19]
[4, 11, 19, 75, 96, 94, 93, 57, 36, 84]
[4, 11, 19, 36, 96, 94, 93, 57, 75, 84]
[4, 11, 19, 36, 57, 94, 93, 96, 75, 84]
[4, 11, 19, 36, 57, 75, 93, 96, 94, 84]
[4, 11, 19, 36, 57, 75, 84, 96, 94, 93]
[4, 11, 19, 36, 57, 75, 84, 93, 94, 96]
[4, 11, 19, 36, 57, 75, 84, 93, 94, 96]
[4, 11, 19, 36, 57, 75, 84, 93, 94, 96]

[33, 80, 8, 68, 58, 17, 58, 91, 89, 77]
[33, 8, 68, 58, 17, 58, 80, 89, 77, 91]
[8, 33, 58, 17, 58, 68, 80, 77, 89, 91]
[8, 33, 17, 58, 58, 68, 77, 80, 89, 91]
[8, 17, 33, 58, 58, 68, 77, 80, 89, 91]
[8, 17, 33, 58, 58, 68, 77, 80, 89, 91]

[28, 0, 74, 73, 29, 11, 69, 7, 80, 97]
[0, 28, 73, 29, 11, 69, 7, 74, 80, 97]
[0, 28, 29, 11, 69, 7, 73, 74, 80, 97]
[0, 28, 11, 29, 7, 69, 73, 74, 80, 97]
[0, 11, 28, 7, 29, 69, 73, 74, 80, 97]
[0, 11, 7, 28, 29, 69, 73, 74, 80, 97]
[0, 7, 11, 28, 29, 69, 73, 74, 80, 97]
[0, 7, 11, 28, 29, 69, 73, 74, 80, 97]

[53, 11, 58, 97, 48, 40, 91, 56, 98, 3]
[53, 11, 58, 97, 48, 40, 91, 56, 98, 3]
[11, 53, 58, 97, 48, 40, 91, 56, 98, 3]
[11, 53, 58, 97, 48, 40, 91, 56, 98, 3]
(continued)
[11, 53, 58, 97, 48, 40, 91, 56, 98, 3]
[11, 48, 53, 58, 97, 40, 91, 56, 98, 3]
[11, 40, 48, 53, 58, 97, 91, 56, 98, 3]
[11, 40, 48, 53, 58, 91, 97, 56, 98, 3]
[11, 40, 48, 53, 56, 58, 91, 97, 98, 3]
[11, 40, 48, 53, 56, 58, 91, 97, 98, 3]
[3, 11, 40, 48, 53, 56, 58, 91, 97, 98]

[51, 16, 29, 81, 24, 46, 47, 5, 10, 86]
[51, 16, 29, 81, 24, 46, 47, 5, 10, 86]
[16, 51, 29, 81, 24, 46, 47, 5, 10, 86]
[16, 29, 51, 81, 24, 46, 47, 5, 10, 86]
[16, 29, 51, 81, 24, 46, 47, 5, 10, 86]
[16, 24, 29, 51, 81, 46, 47, 5, 10, 86]
[16, 24, 29, 46, 51, 81, 47, 5, 10, 86]
[16, 24, 29, 46, 47, 51, 81, 5, 10, 86]
[5, 16, 24, 29, 46, 47, 51, 81, 10, 86]
[5, 10, 16, 24, 29, 46, 47, 51, 81, 86]
[5, 10, 16, 24, 29, 46, 47, 51, 81, 86]

[38, 33, 51, 89, 57, 95, 72, 23, 80, 78]
[23, 33, 51, 89, 57, 95, 72, 38, 80, 78]
[23, 33, 51, 89, 57, 95, 72, 38, 80, 78]
[23, 33, 38, 89, 57, 95, 72, 51, 80, 78]
[23, 33, 38, 51, 57, 95, 72, 89, 80, 78]
[23, 33, 38, 51, 57, 95, 72, 89, 80, 78]
[23, 33, 38, 51, 57, 72, 95, 89, 80, 78]
[23, 33, 38, 51, 57, 72, 78, 89, 80, 95]
[23, 33, 38, 51, 57, 72, 78, 80, 89, 95]
[23, 33, 38, 51, 57, 72, 78, 80, 89, 95]
[23, 33, 38, 51, 57, 72, 78, 80, 89, 95]
```

Below is given a list an one intermediate step in a sort algorithm. Determine the algorithm (or more than 1 if it is ambiguous)

```
[43, 76, 32, 76, 15, 90, 33, 0, 82, 47]
[0, 15, 32, 33, 43, 47, 76, 76, 82, 90]

[37, 30, 87, 41, 66, 61, 20, 80, 56, 17]
[20, 30, 37, 17, 41, 56, 61, 66, 80, 87]

[4, 74, 73, 98, 37, 45, 99, 62, 79, 77]
[4, 37, 45, 73, 74, 98, 99, 62, 79, 77]

[41, 94, 24, 14, 27, 11, 3, 1, 54, 14]
[1, 3, 11, 14, 14, 24, 94, 41, 54, 27]

[66, 74, 41, 14, 86, 94, 71, 10, 49, 9]
[14, 41, 66, 71, 74, 86, 94, 10, 49, 9]

[60, 18, 62, 72, 79, 39, 97, 6, 64, 30]
[18, 60, 62, 72, 79, 39, 97, 6, 64, 30]

[86, 60, 90, 79, 57, 73, 60, 53, 35, 30]
[60, 79, 86, 90, 57, 73, 60, 53, 35, 30]

[47, 28, 60, 65, 65, 89, 78, 33, 74, 26]
[26, 28, 33, 47, 60, 89, 78, 65, 74, 65]
```

KEY

```
int[] a = newArray(10, 0, 100);

a = insertionSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = selectionSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = selectionSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = bubbleSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = bubbleSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = insertionSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = insertionSort(a);
System.out.println();

a = newArray(10, 0, 100);
a = selectionSort(a);
System.out.println();
```

```
int[] a = newArray(10, 0, 100);
        a = selectionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = bubbleSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = insertionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = selectionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = insertionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = insertionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = insertionSort(a);
        System.out.println();

        a = newArray(10, 0, 100);
        a = selectionSort(a);
        System.out.println();
```