c. bubble-sort

```
1,      for i = 0 to A.length-1
2,              for j= A.length downto i+1
3,                      if A[j]<A[j-1]:
4,                              exchange A[j] with A[j-1]
```

Loop invariants:

Initialization:

For the first iteration, i=0, then we go into the second loop in lines through 4 find the smallest element of the whole array A, it kinds of like make the smaller of two elements stand out, and then move the smaller element to first position and the bigger one to the later position, after first iteration, we get the smallest element of the whole array.

Maintain:

The for loop in lines 2 through 4 maintains the following loop invariant: At the start of each iteration, the position of the smallest element of A[i..n] is at most j. This is clearly true prior to the first iteration because the position of any element is at most A.length. To see that each iteration maintains the loop invariant, suppose that j = k and the position of the smallest element of A[i..n] is at most k. Then we compare A[k] to A[k − 1]. If A[k] < A[k − 1] then A[k − 1] is not the smallest element of A[i..n], so when we swap A[k] and A[k − 1] we know that the smallest element of A[i..n] must occur in the first k − 1 positions of the subarray, the maintaining the invariant. On the other hand, if A[k] ≥ A[k − 1] then the smallest element can't be A[k]. Since we do nothing, we conclude that the smallest element has position at most k − 1. Upon termination, the smallest element of A[i..n] is in position i.

Termination:

The for loop in lines 1 through 4 maintain the following loop invariant: At the start of each iteration the subarray A[1..i − 1] contains the i − 1 smallest elements of A in sorted order. Prior to the first iteration i = 1, and the first 0 elements of A are trivially sorted. To see that each iteration maintains the loop invariant, fix i and suppose that A[1..i − 1] contains the i − 1 smallest elements of A in sorted order. Then we run the loop in lines 2 through 4. We showed in part b that when this loop terminates, the smallest element of A[i..n] is in position i. Since the i − 1 smallest elements of A are already in A[1..i − 1], A[i] must be the i th smallest element of A. Therefore A[1..i] contains the i smallest elements of A in sorted order, maintaining the loop invariant. Upon termination, A[1..n] contains the n elements of A in sorted order as desired.