Consider the algorithm for the sorting problem that sorts an array by counting, for each of its elements, the number of smaller elements and then uses this information to put the element in its appropriate position in the sorted array:

```
ALGORITHM  ComparisonCountingSort($A[0..n-1]$)
    //Sorts an array by comparison counting
    //Input: Array $A[0..n-1]$ of orderable values
    //Output: Array $S[0..n-1]$ of $A$'s elements sorted
    //    in nondecreasing order
    for $i \leftarrow 0$ to $n-1$ do
        $Count[i] \leftarrow 0$
    for $i \leftarrow 0$ to $n-2$ do
        for $j \leftarrow i+1$ to $n-1$ do
            if $A[i] < A[j]$
                $Count[j] \leftarrow Count[j]+1$
            else $Count[i] \leftarrow Count[i]+1$
    for $i \leftarrow 0$ to $n-1$ do
        $S[Count[i]] \leftarrow A[i]$
    return $S$
```

a. Apply this algorithm to sorting the list 60, 35, 81, 98, 14, 47.

b. Is this algorithm stable?

c. Is it in-place?