获得的答案 **返回**

COUNTING-SORT is stable:

• It is known that a sorting algorithm is table, if any two elements with same value appear in the same order in sorted array, as they appear in the original array.

• COUNTING-SORT is stable, as the input order among equal elements in the input array is preserved in the sorted array. That is, the elements with same value appear in the same order as they appear the original array.

Proving that C OUNTING -S ORT is stable:

• The final for loop of COUNTING-SORT runs from *A.length* to 1 and after *i* is stored in *B*, *C*[*i*] is decreased by 1. Thus, if there are any two elements with same value in *A*, the element with higher index in *A* is appear after the element with lower index in *A*.

• For example, consider two equal elements in *A*, stored at indices i_1 and i_2, where $i_1 < i_2$. That is, $A[i_1] = A[i_2]$.

• If there are *k* number of elements less than or equal to $A[i_1]$ or $A[i_2]$. That is, $C[A[i_1]]$ or $C[A[i_2]]=k$.

• Now, the final for loop processes $A[i_2]$ first. That is, $A[i_2]$ is stored in *B* at index *k* and $C[A[i_1]]$ is decreased by 1. That is, $C[A[i_1]]=k-1$.

• Later $A[i_1]$ is stored in *B* at index *k*-1. Obviously, $k-1 < k$. Thus, $A[i_1]$ and $A[i_2]$ appear in the same order in *B* as they appear in *A*.

Since the COUNTING-SORT is preserving the order of the elements with the same value, COUNTING-SORT is stable.

Example:

Even after applying COUNTING-SORT, the order of *A*[1] and *A*[4] is same in B.