Problem 3

a. First, Prove a is empty, the algorithm terminate.

If a is empty, will move b one by one until b empty, so algorithm return C, terminated.

The same goes with b is empty, the algorithm terminate.

Second, assume length of $a \le n_1$, length of $b \le n_2$, the algorithm terminate

Third, we need prove length of $a = n_1+1$, length of $b = n_2+1$, the algorithm terminate

① if the first element of a and b be moved alternately during the algorithm, there must be a moment when both a and b are moved at least one element. Thus, length of $a \le n_1$, length of $b \le n_2$, according to Second assumption. Proved.

② if the first element of a always less than b, the algorithm will move a one by one until a is empty, according to first prove, algorithm will terminate.

③ if the first element of b always less than a, same reason with ②.

In conclusion, the merge algorithm will terminate.

b. Notation: A, B is the set of elements in array a, b, C is the set of elements in output.

Assume, exist one element X in A that C doesn't contain X, which means the algorithm does not move X to C.

However, in execution of the algorithm, when the length of a is not empty (at least X exist). ① If b is empty, A will be moved one by one, so eventually C will contain X. Contradict with assumption.

② if b is not empty, if exist $b_i > X$, X will be moved to C before b_i. So C will contain X. if all $b < X$, it will be the same situation in ①.

In conclusion, doesn't exist one element in A that not in C. We can prove this in B same way. So The output array contains union of elements of input array.

C. Same Notation with b.

First prove, if randomly picked two elements from C, they are in correct sorted order. If this is proved, we can have

$C_1 \le C_2$, $C_2 \le C_3$, $\cdots$ $C_{n-2} \le C_{n-1}$, $C_{n-1} \le C_n$.

So $\therefore$ $C_1 \le C_2 \le C_3 \le \cdots \le C_{n-1} \le C_n$. The problem is proved.

So assume randomly pick C_i, C_j from C. $(1 \le i < j \le n.)$

① if C_i, C_j both from A or B. as A or B is sorted,

if $i<j$. $C_i \le C_j$. they are in correct order.

② if One of C_i, C_j is from A, the other from B.

because $i<j$ $\therefore$ there exist a moment C_i just moved to C while C_j still in array B. ① if C_j is the first element of B now, $C_i < C_j$ (according to algorithm) ② if C_j is not the first of B, must exist C_k $(k<j)$ and C_k is the first element of B. So $C_i < C_k$, while $C_k < C_j$ as B is sorted, thus $C_i < C_j$. they are in correct order.

In conclusion, we randomly pick two elements from C, they must be in correct order.

So if intput are in order, the output is also in order.

(~~I only proved the situation where A and B are in an increasing~~ Order, if they are reversed, the conclusion cannot be proved)