a. There is nothing else we need to prove. Proving these two is sufficient to show that the program correctly sorts.

b. Loop Invariant: The loop invariant of lines 2-4 is that after each loop iteration, the smallest element in the sub-array $A[i : n]$ is present in the sub-array $A[i : j-1]$.

Initialisation: This is true before the 1^{st} or after the 0^{th} iteration, where we can say $j = n+1$. In this case, the smallest element of sub-array $A[i : n]$ is obviously present in $A[i : j-1]$, because $A[i : j-1] = A[1 : n]$.

Maintenance: Let us say that before the iteration, we know $A[i : j]$ contains the smallest element of sub-array $A[i : n]$, then there we need to analyse three possibilities.

1. The smallest element is in index j - In this case, since $A[j]$ is necessarily smaller than $A[j-1]$, thus the swap will happen, and the new position of will be $j-1$. (There is also the possibility where they are equal, in which the smallest element is in index $j-1$ both before and after the loop.)

2. The smallest element is in index $j-1$ - In this case, since $A[j-1]$ is necessarily smaller than or equal to $A[j]$, the swap will not happen, and its position after the loop will be $j-1$.

3. The smallest element is in $A[i : j-2]$ - In this case, since the position of the smallest element cannot get affected because the comparison and exchange is only between $A[j]$ and $A[j-1]$

In each of these cases, the smallest element of the sub-array is present in $A[1 : j-1]$ after the loop.

Termination: The loop terminates after the loop in which $j = i+1$, thus after the loop terminates, the smallest element of $A[i : n]$ is present in $A[i : i]$ or in the index i.

c. Loop Invariant: After the i^{th} iteration, the elements $A[1 : i]$ are in sorted order, and all the elements in $A[i+1 : n]$ are greater than or equal to $A[i]$.

Initialisation: After the 1^{st} iteration, the smallest element of $A[1 : n]$ is in $A[1]$ (by the termination condition of the for loop in lines 2-4. Thus, $A[1 : 1]$ is in sorted order (since there is only one element) and the elements in $A[2 : n]$ are all greater than or equal to $A[1]$.

Maintenance: Let us say before the i^{th} iteration, $A[1 : i-1]$ are in sorted order, and all elements in $A[i : n]$ are greater than or equal to $A[i-1]$. After the loop, by the termination condition of the for loop of lines 2-4, $A[i]$ holds the smallest element of $A[i : n]$. Thus we have $A[i]$ greater than equal to all the elements in $A[1 : i-1]$. Since $A[i : i-1]$ is in sorted order, we therefore have $A[1 : i]$ in sorted order. And since $A[i]$ contains the smallest element of $A[i : n]$, all the elements in $A[i+1 : n]$ are greater than or equal to $A[i]$, satisfying our loop invariant.

Termination: The loop ends after the $(n-1)^{th}$ iteration, which means at the end, we have $A[1 : n-1]$ in sorted order, and all the elements of $A[n : n] =$

$A[n]$ greater than or equal to $A[n-1]$. However, this just means $A[1:n]$ is in sorted order. Thus the inequality is proven.

d. The worst-case running time of both is $\Theta(n^2)$