

proof of invariant:

Initialization: Before the loop starts when i comes at 0th iteration there is there is only one element in the sub-array which itself is sorted.

Maintenance: Here in the inner loop in each iteration the if statement is responsible for comparing $A[j]$ and $A[j-1]$. And that if $A[j]$ is smaller than $A[j-1]$ swap occurs.

At each iteration the smallest element will move to the left and this is due to the inner for loops. Thus by the end of first iteration of the outer loop we will have the first smallest element in the array at position $A[1]$.

Similarly, after second iteration of outer for loop the second smallest item will appear to the left side of the array which is position $A[2]$.

Permutation: After each finite iteration of inner loops, followed by the outer loop at each position in the array:

$$A[n] < A[n+1] \rightarrow []$$

True [] showing a valid sorted list of elements in the array.

True Complexity: $O(n^2)$

Outer loop runs for n times.

For each k^{th} iteration of outer loop, the inner loops run k times.

So, overall,

$$[] > []$$

$$\text{True Complexity} = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2}$$

$$= O(n^2)$$