**Loop Invariant:**

At the start of each iteration of the outer loop (indexed by j), the subarray A[1] to A[j-1] is sorted, and the element A[j] is the next one to be inserted into its correct position by the inner loop (indexed by i).

**Breakdown:**

**Initialization (before the outer loop starts)**: When j = 2, the subarray A[1] is trivially sorted because it consists of only one element.

**Maintenance (during each iteration of the outer loop):**The outer loop starts at j = 2 and progresses to n. At each step, the subarray A[1] to A[j-1] is sorted.

The inner loop (indexed by i ) moves backward from j to 1 (or until the correct position is found) to insert the element A[j] into its correct position in the sorted subarray.

After A[j] is inserted in its correct position, the subarray A[1] to A[j-1] is sorted, maintaining the invariant for the next iteration ofj.

**Termination (when the outer loop ends):** When the outer loop finishes at j = n+1, the subarray A[1] to A[j-1] (i.e., the entire array) is sorted, which satisfies the problem’s goal.

This loop invariant explains that at each step of j, the element A[j] is inserted into the sorted subarray A[1] to A[j-1] using the inner loop controlled by i.