2

a) The all elements in A also exist in A’, though they may be in a different order

b) Loop Invariant 1) The subarray A[j...n] consists of elements of A[j...n] from before the loop, possibly in a different order

2) A[j] is the smallest of those elements

Initialization: Must hold, because there is only one element and it is the very last element in the array

Maintenance: (1) holds because at each step, we replace A[j] with A[j-1], and we’re only adding the previous element and possibly swapping two values. (2) since the loop invariant states that A[j] is the smallest of A[j...n] and A[j-1] becomes the smallest of A[j] and A[j-1]

Termination: When the loop terminates, j=i, which implies the A[i] is the smallest element of the subarray A[i...n] and contains the original elements in the same order

c) Loop Invariant: At the beginning of each iteration, A[1...i-1] consists of sorted elements, all of which are less than or equal to the ones in A[i...n]

Initialization: Array is empty initially

Maintenance: Inner loop invariant states that at each iteration, A[i] becomes the smallest element of A[i...n], while the rest get shuffled around. At the end of the loop, A[i] < A[k], for i<k

Termination: At termination, i=n, where n is the length of the array. If we substitute I for n in the loop invariant, we can state the A[1...n] consists of the original elements, but they are sorted. At termination, this is the entire array, so the whole array is sorted.

d) θ(n2), which is the same as Insertion Sort

e)

i. If every other element is out of order, such as [1, 3, 2, 4]

ii. There is no permutation where Insertion Sort takes θ(n2), but Bubble Sort takes θ(n)