Correctnes of Selection sort: (By loop invariant)

At the beginning of the ith iteration of outer loop, the Subarray contains i-1 elements of the original array in sorted order i.e; A[1,.... i-1] in sorted order

Initialization:
Before the first iteration where i=1, we have the subarray A[1,.. - i-1] is empty. The loop invariant holds tivially, before the first iteration of the loop.

Maintenance: Now, we tackle the second property: Showing that each iteration maintains the loop invariant. Let's assume that our loop invariant holds at the beginning of the ith iteration.

⇒ A[1 ... i-1] contains the i-1 elements in sorted order.

⇒ In the ith- iteration:

- The inner loop finds the minimum element index in the unsorted portion.
- And the minimum element is swapped with arr[i] and now A[1 ... i] contains the i elements in sorted order and the loop invariant is maintained.

Termination: Now, we examine what happens when the loops terminates

⇒ The outer loop terminates when i reaches n where n is length of array

⇒ The loop invariant states that A[1... n] elements are sorted which represent our entire array.

Hence the algorithm is correct.