Selection Sort Correctness

Pseudocode:

```
for i = 1 to length(Array)-1;
          minElement = i
           for j = i+1 to length(Array);
              if Array[j] < Array[minElement]
                   minElement = j
            temp = Array[i]
            Array[i] = Array[minElement]
            Array[minElement] = temp
```

Inner Loop:

Loop Invariant: start of each iteration, Array[minElement] is less than or equal to elements in the subarray Array[i..j]. After first iteration i, the i min elements of Array are in sorted order in positions 1 to i.

Initialization: Before the first iteration of the inner loop, j = i, therefore the subarray to consider is Array[i..i] or simply Array[i] Previously minElement was set to the index I, minElement indexes the only element in Array[i].

Maintenance: Before iteration j+1, we assume that minElement is the index of the smallest element in subarray A[i..j]. During iteration j+1, two cases :

1. Array[j+1] < Array[minElement]

2. Array[j+1] >= Array[minElement].

When the inner loop completes, minElement refers to the index of the smallest element in Array[i+1..n]. Since Array[1..i] consists of the i smallest elements, minElement is the i+1st smallest element. We move this to Array[i+1] via the swap. After the swap, and at the end of the iteration Array[1..i] consists of the i smallest elements in positions 1 to i.

Termination: At the end of the inner loop, minElement is the index of an element less than or equal to all elements in Array[i..j]. So minElement is the index of the smallest element in Array[i..n]. When the outer loop terminates, i is the length of the array. Therefore we can conclude that Array[1 ..n] has all elements of Array, sorted order.