2. Argue the correctness of Selection Sort Algorithm

```
for int i = 0 to A.length-1 {
    min = i
    for j = i+1 to n {
        if A[j]<A[min] {
            min = j
        }
    }
    swap(A[min],A[i])
}
```

i. Selection Sort divides the array into sorted and unsorted parts.

ii. We find the minimum element from the unsorted part and swap it with the first element of the unsorted part if it is smaller.

iii. We keep adding elements in the sorted part until the entire array is sorted.

Loop Invariant – At the beginning of each iteration i, the first i elements are sorted and contain the smallest i elements in their correct positions

Proof of Induction:

i. Base Case
Before the first iteration (i=0), the sorted part of the array is empty. So, the invariant holds true here.

ii. Step Case
Before iteration i, i elements are sorted correctly. Selection sort algorithm finds the smallest element in the unsorted part. It swaps this minimum element with A[i], ensuring that the smallest i+1 elements are correctly placed and sorted. The next iteration begins at i+1 and the invariants holds true in the case.

iii. Termination
The loop terminates when i=n-1. Here, the first n-1 elements are already sorted and only one element is left and it is in the correct position.
Hence, the entire array is sorted. This proves the correctness of Selection Sort.

Best, Worst and Average case time complexity – O(n^2)