DS 320 Algorithms for Data Science **Lecture #2 Worksheet**
Spring 2026 Prof. Kira Goldner and Jeffrey Considine

Insertion Sort and Induction

We will analyze the runtime of the following algorithm.

Algorithm 1 InsertionSort(A).

```
**Input:** $A$ is an array of integers. It is indexed 1 to $n$.
for $i = 2$ to $A.length$ do
    $current = A[i]$
    $j = i - 1$
    while $j > 0$ and $A[j] > current$ do
        $A[j+1] = A[j]$
        $j = j - 1$
    end while
    $A[j+1] = current$
end for
return $A$
```

First, we analyze the algorithm's runtime. This time, we'll *formally* argue its runtime.

Theorem 1. *Insertion Sort runs in time $O(n^2)$.*

Proof.

Now, we argue the algorithm's *correctness*—that is, that on *every possible input*, it correctly outputs a sorted version.

Correctness via Induction

Theorem 2. *For any input instance A, Insertion Sort returns an array sorted in ascending order.*

We want to use induction on a claim that, when true for all integers, proves our claim. What claim might that be?

Proof. We show the following by induction on i: (*premise*)

Base Case:

Inductive Hypothesis:

Inductive Step:

We need another component to prove the inductive step. Separately, we want to prove the following:

Lemma 1. (Loop Invariant) *The "while" loop shifts $A[j+1, i-1]$ to $A[j+2, i]$ in the same order for some j.*

□

Loop Invariants

We'll use a different technique now to prove Lemma 1.

Definition 1. A *loop invariant* is something that is true before we start and after every iteration of a loop.

We prove that a loop invariant is true by showing the following three things about it:

- **Initialization:** It is true prior to the first iteration of the loop.
- **Maintenance:** If it is true before an iteration of the loop, it remains true before the next iteration.
- **Termination:** When the loop terminates, the invariant gives us a useful property that helps show that the algorithm is correct.

See CLRS Section 2.1 for details, p.18 in the Third Edition. We use this to prove our Lemma.

Proof of Lemma 1. We will prove this formally as a loop invariant.

Initialization: Before the first iteration of the "while" loop,

Maintenance: If our statement holds before an iteration of the loop—that $A[j+1, i-1]$ is shifted to $A[j+2, i]$ in order—then

Termination: When the loop terminates,

□