

EECS3101 (Section E) Summer 2026
Tutorial 7
Tracing Loop Invariant, Proving Algorithm Correctness
Chen-Wei Wang

Release Date: Thursday, July 2
No Submission Required: Complete for Learning & Test Prep

Contents

1	Exercise: Proving Algorithm Correctness using Loop Invariant	2
---	--	---

1 Exercise: Proving Algorithm Correctness using Loop Invariant

Consider the following linear-time, iterative algorithm, which intends to find the maximum value from the input array:

```
1 int findMax (int[] a) {
2   assert Q; /* precondition satisfied? */
3   int i = 0;
4   int result = a[i];
5   assert LI; /* invariant established? */
6   while(i != a.length) {
7     if(a[i] > result) {
8       result = a[i];
9     }
10    i = i + 1;
11    assert LI; /* invariant preserved? */
12  }
13  assert R; /* postcondition satisfied? */
14  return result;
15 }
```

1. Specify the precondition (Q) and postcondition (R) of the above algorithm.
2. Can the following Boolean condition serve as an appropriate loop invariant?

$$\forall j \bullet 0 \leq j \leq i \Rightarrow \text{result} \geq a[j]$$

Justify your answer by tracing through how the above Boolean condition is checked as the loop invariant.

3. If the above Boolean condition is not an inappropriate loop invariant, suggest an appropriate one.
4. Given that the appropriate loop invariant is established and preserved, prove that the above algorithm is *correct*.