Proving Termination and Analyzing the Running Time of a Simple Algorithm with a While Loop

A Sample Assignment

Consider the following computation problem.

Is Array Increasing?

Precondition: An integer array A with some positive length n is given as input.

Postcondition: The Boolean value `true` is returned if $\mathtt{A}][i] < \mathtt{A}[i+1]$ for every integer i such that $0 \le i \le n-2$. The Boolean value `false` is returned, otherwise.

Consider, as well, the following algorithm.

```
boolean arrayIncreasing ( integer[] A ) {
  integer i := 0
  while (i $\le$ A.length $-$ 2) {
    if (A[i] $\ge$ A[i $+$ 1]) {
      return false
    } else {
      i := i $+$ 1
    }
  }
  return true
}
```

1. State a ***bound function*** for the `while` loop in this algorithm, and prove that your answer is correct.

2. Use this to prove that an execution of this algorithm always terminates if the execution begins with the precondition for the “Is Array Increasing?” problem satisfied.

   If you completed the Assignment for Reading #3 then you have also proved that this algorithm is partially correct. Note that it now follows that this correctly solves the “Is Array Increasing?” problem.

3. State an upper bound (and, if you can, a lower bound) for the number of steps carried out by this algorithm when it is executed with an input array of positive length n.

   Your bounds should be functions of n. While intermediate work might include summations, your final bounds should not include any.