COSC 290, Fall 2017 Problem set # 9: Induction Proofs Prof. M. Hay

Name: ______________________________

Today in class, you are asked to study the following algorithms and prove claims about them.[1] You are encouraged to work with others, but you must write up your own answer. Please turn this in at the end of class.

```
procedure BinarySearch(A, n, x)                ▷ Find x in sorted array A of length n
    Set l to 0 and u to n − 1.
    while l < u do
        Set m to ⌊(l + u)/2⌋.                   ▷ integer division
        if A[m] = x then
            Set l = u = m.
        else if A[m] > x then
            Set u to m.
        else
            Set l to m.
        end if
    end while
    If A[l] = x return l else return -1.
end procedure
procedure BinarySearch2(A, n, x)               ▷ Find x in sorted array A of length n
    Set l to 0 and u to n − 1.
    while True do
        if l > u then
            Set i to -1 and break.
        end if
        Set m to ⌊(l + u)/2⌋.                   ▷ integer division
        if x < A[m] then
            Set u to m − 1.
        else if x > A[m] then
            Set l to m + 1.
        else                                    ▷ It must be that A[m] = x
            Set i to m and l = u = m and break.
        end if
    end while
    return i
end procedure
```

Decide whether either of the above algorithms is a correct implementation of binary search. Prove your claims. If you think the algorithm has a bug, provide a counter example (an input where the program does not return the correct answer). If you think the algorithm is correct, write a proof with two parts:

- Claim 1: If the algorithm terminates, it returns the correct answer.
- Claim 2: The algorithm eventually terminates.

[1] This exercise adapted from Bentley's *Programming Pearls*.

Here's a suggested strategy: use induction on t, the number of iterations of the while loop, to prove the following lemma.

Lemma 1. *The following* invariant *holds throughout the execution of the algorithm: If x is in array A, then the set $\{l, \ldots, u\}$ contains the index where x is located. (If $l > u$, this is an empty set and thus the set does not contain any index.)*

You can use MustBe(a, b) as shorthand for "if x is in array, the set $\{a, \ldots, b\}$ contains its index. Similarly CantBe(a, b) is shorthand for "if x is in array, its index is *not* contained in the set $\{a, \ldots, b\}$."

After proving Lemma 1, show how you can use it to prove Claim 1. For Claim 2, you might use the lemma but in a different way (hint: what can you say about the cardinality of the set $\{l, \ldots, u\}$?)

1. Write your proofs here.