NOTES ON PROVING CORRECTNESS OF BINARY SEARCH

We wish to prove that the program given below is correct with respect to the following *Precondition* and *Postcondition.* It is assumed throughout that x, n, *first*, *last* and *mid* always contain integer values, that *found* holds a Boolean value, and that A(u) holds an integer value for every integer u.

Precondition: $n \geq 1$. For all u, v such that $1 \leq u \leq v \leq n$, $A(u) \leq A(v)$.

Postcondition: If for some u, $1 \leq u \leq n$, $A(u)=x$, then *found*=true and $1 \leq first \leq n$ and $A(first)=x$. If there is no u, $1 \leq u \leq n$, such that $A(u)=x$, then *found*=false.

```
first:=1
last:=n
loop
        exit when first=last
        mid:= (first+last) div 2
        if x>A(mid) then
                first:=mid+1
        else
                last:=mid
        end if
end loop
if x=A(first) then found:=true else found:=false end if
```

From now on, we will assume that n and A are fixed, and satisfy the Precondition. To prove correctness for this algorithm, the key lemma to be proved is as follows.

Loop Invariant Lemma: At every visit to the exit test

(1) $1 \leq first \leq last \leq n$ and

(2) if there is some u, $1 \leq u \leq n$, $A(u)=x$, then there is some u, $first \leq u \leq last$, $A(u)=x$.

A key point which is needed to prove this lemma is the following sub-lemma, which should be proved separately.

Sub-Lemma:
For all integers a and b, if $a<b$, then $a \leq (a+b) \text{ div } 2 < b$.

Proof of Sub-Lemma:
Since $a \leq b$, $(a+b) \text{ div } 2 \geq (a+a) \text{ div } 2 = a$.
Since $a \leq b-1$, $(a+b) \text{ div } 2 \leq (b-1+b) \text{ div } 2 = b-1 < b$.

Proof of Loop Invariant Lemma:
For every i such that the loop gets executed at least i times, let $first_i$ and $last_i$ be the values of *first* and *last* after the *i-th* execution.
Let P(k) be: If the loop is executed at least k times, then

(1) $1 \le first_k \le last_k \le n$ and

(2) if there is some u, $1 \le u \le n$, A(u)=x, then there is some u, $first_k \le u \le last_k$, A(u)=x.

We will prove that P(k) holds for all natural numbers k, by (simple) induction.

Base Case: We have to show that P(0) holds. This is left as an exercise.

Induction Step: Let $i \ge 0$ and assume P(i) holds. We want to prove P(i+1). Assume the loop gets executed at least i+1 times. From P(i) we know $1 \le first_i \le last_i \le n$, and since the program didn't halt after i iterations, $first_i \neq last_i$; so we know $1 \le first_i < last_i \le n$. Because of the way *mid* gets assigned, the sub-lemma tells us that $1 \le first_i \le mid < last_i \le n$.

CASE 1: x>A(*mid*).

Then $first_{i+1}$=mid+1 and $last_{i+1}$=$last_i$. So $1 \le first_i < first_{i+1} \le last_{i+1} = last_i \le n$, so $1 \le first_{i+1} \le last_{i+1} \le n$. Now assume that A($u$)=$x$ for some u, $1 \le u \le n$. By P(i), A(u)=x for some u, $first_i \le u \le last_i$. Because A is sorted between positions 1 and n, it is sorted between positions $first_i$ and $last_i$; since $first_i \le mid < last_i$ and x>A(*mid*), it must be the case that A(u)=x for some u, $mid+1 \le u \le last_i$, that is for some u, $first_{i+1} \le u \le last_{i+1}$.

CASE 2: $x \le$A(*mid*). This case is left as an exercise.

Proof of Partial Correctness: Assume that the program halts. So *first*=*last*, and so by the Loop Invariant Lemma, $1 \le first = last \le n$.

CASE 1: There is some u, $1 \le u \le n$, A(u)=x.

By the Loop Invariant Lemma, there is some u, $first \le u \le last$, A(u)=x.
So A(*first*)=x, *found* gets assigned true, and the Postcondition holds.

CASE 2: Otherwise. This case is left as an exercise.

Proof of Termination: Consider the integer quantity $last_i - first_i$. By the Loop Invariant Lemma, this quantity is always ≥0. So it suffices to show that $last_{i+1} - first_{i+1} < last_i - first_i$ (assuming there are at least i+1 iterations). So consider an arbitrary $i \ge 0$, and let $mid = (first_i + last_i)$ div 2.

CASE 1: x>A(*mid*).

From the proof above, we have $1 \le first_i < first_{i+1} \le last_{i+1} = last_i \le n$, and so $last_{i+1} - first_{i+1} < last_i - first_i$.

CASE 2: $x \le$A(*mid*). This case is left as an exercise.