Michael Yangmi & Joe Webster

1) $\forall i \forall j : (1 \le i < j \le n) \Rightarrow (A[i] \le A[j])$
$\exists i : (1 \le i \le n) \wedge A[i] = x$

2.1) Given that the arrays starting index is "1", if "lo" is assigned "1", then "hi" should be assigned "n", or the size/length of the array. If hi was assigned "n-1", and lo was assigned "1", we would not be looking at the entire array to sort it. For instance:

1	2	3	4	5	6
12	15	17	17	26	30
$\downarrow$				$\downarrow$	
$lo \leftarrow 1$				$hi \leftarrow n-1$	

$\therefore$ lo needs to be 1, and hi needs to be n.

2.2) $lo \leftarrow 1$, $hi \leftarrow$ "n"; where 'n' is the total number of elements in A.

3.1) If at 'T', $hi \leftarrow q$, then we cannot maintain Φ. This is because 'x' will exist outside of the now established range. Therefore, in this case, it will be impossible to hold postulate (3). 'x' will not occur within the newly formed $A[lo...hi]$.

3.2.A) In the case where $lo \leftarrow q$ at 'T', and $hi \leftarrow q$ at 'E', (3) of Φ is maintained because:

- If $A[q] < x$, we assign q to lo. This creates a new range to work with where $lo' \leftarrow q$ (new lo) and $q' \leftarrow ((lo' + hi)/2)$.
  - ↳ This means 'x' must be/exist in $A[lo' ... hi]$ for any given case where $A[i] < x$.

- If $A[q] > x$, we assign q to hi. This means that 'x' exists between lo and q. So we reassign $hi' \leftarrow q$ and $q' \leftarrow ((lo + hi')/2)$
  - ↳ This means 'x' must exist in $A[lo ... hi']$ for any given case where $A[i] < x$.

Michael Yaingni & Joseph Webster

3.2.B) An example would be where the array has three non-decreasing elements defined as:

$A[\,] = [1, 3, 5]$

$n = 3$

$x = 5$ // value we are looking for

$lo = 1$

$hi = 3$

$q = 2$

- This scenario would cause an infinite loop because q would be staying in the same spot.
  - ↳ This happens because q is calculated by taking $((lo + hi)/2)$. This is integer division and will take 2 from $\frac{5}{2} = 2.5$, lo will continually be assigned the value of q, and q will always be 2.

3.3) $\boxed{T}$

lo ← q + 1

$\boxed{E}$

hi ← q − 1

4) Given that an array 'A', and a value to look for 'x' are declared globally.

```
function FIND(low, high)
    q ← (low+high)/2
    if (A[q] != x)
        if (A[q] < x)
            low = q + 1
        else
            high = q - 1
        return FIND(low, high)
    return q
```