Algorithm FindMin \( (A, n) \)

Input: Array \( A \) of \( n \) integers

Output: Return the smallest element in \( A \)

\[
\begin{align*}
& i \leftarrow 1 \\
& m \leftarrow A[0] \\
& \text{while } (i < n) \text{ do } \\
& \quad \text{if } (A[i] < m) \text{ then } m \leftarrow A[i] \\
& \quad i \leftarrow i + 1 \\
& \text{return } m
\end{align*}
\]

Loop invariant: \( \text{At iteration } i, m = \min \{A[0], \ldots, A[i-1] \} \)

Goal: Prove the loop invariant holds

1. Initialization: Before the start of the loop, \( \text{L.I.} \) holds

\[
\begin{align*}
& i = 1, \ m = A[0] = \min \{A[0], \ldots, A[0] \} \quad \text{if } i = 1
\end{align*}
\]

2. Maintenance: Assume \( \text{L.I.} \) holds at beginning of loop, an iteration of loop

We must show that \( \text{L.I.} \) holds at the end of that iteration

Assume \( m = \min \{A[0], \ldots, A[i-1] \} \)

If \( (A[i] < m) \), then replacing \( m \) with \( A[i] \)

result is \( m = \min \{A[0], \ldots, A[i] \} \)

If \( (A[i] \geq m) \), then \( m \) remains unchanged,

\( m \) is now \( m = \min \{A[0], \ldots, A[i] \} \)

After increasing \( i \) by one: \( m = \min \{A[0], \ldots, A[i-1] \} \)

L.I.
3. Termination

3.1. The algorithm will stop because the counter variable $i$ gets increased by one at each iteration. So, it will eventually reach $n$.

3.2. When loop terminates,

$$m = \min \{ A[i], \ldots, A[n-1] \}$$

Loop stops when $i = n$
Loop invariant says $m = \min \{ A[0], \ldots, A[i-1] \}$

$$\iff m = \min \{ A[0], \ldots, A[n-1] \}$$