CSE 417 25au Homework 0: Sample

Released: N/A

First due by: N/A

Last resubmissions by: N/A

Problem 0: Linear search

Given an array of integers $A[1 \dots n]$ and an integer k, return the first index i such that $A[i] = k$. In other words, $A[i] = k$ and for all $j < i$, we have $A[i] \neq k$. If no such index exists, return "impossible".

Write pseudocode solving the problem, demonstrate the pseudocode on example inputs that encompass the major cases, and provide a full proof of correctness.

Solution

Pseudocode:

```
1: for i = 1, ..., n do
2:     if A[i] = k then
3:         return i
4: return "impossible"
```

Examples: On input $A = [10, 20, 20]$ and $k = 0$, the algorithm would test if $10 = 0$, if $20 = 0$, and again if $20 = 0$, and since each test fails, we return "impossible". For the same input and $k = 20$, we would test if $10 = 20$, and then if $20 = 20$, which passes the test, so we return the index 2.

Correctness: No lines will raise errors, because the only call that could raise errors is array access $A[i]$, but we know $1 \leq i \leq n$ by line 1, so this is okay. The program always terminates because we only have for loops.

To show that the program meets the specification, we note the invariant that at the end of iteration i, we know $A[j] \neq k$ for all $1 \leq j \leq i$.

- **Before the loop**: True because there are no j such that $1 \leq j \leq 0$.
- **After iteration** i: By the claim on the previous iteration, we know $A[j] \neq k$ for all $1 \leq j \leq i - 1$. Since we reached the end of iteration i, the if condition did not fire, so $A[i] \neq k$ as well, which completes the proof.

Because the specification has two main cases, let's consider them separately.

Case 1: Suppose A contains k. It is not possible that we reach the end of iteration n (and return "impossible"), because the loop invariant would imply that $A[j] \neq k$ for all $1 \leq j \leq n$, which is false. So the code terminates early in the middle of some iteration i, satisfying $A[i] = k$, and by the loop invariant we know $A[j] \neq k$ for all $1 \leq j \leq k$, thus returning i is correct.

Case 2: Suppose A does not contain k. It is not possible that we return early, because that means some $A[i] = k$ check needs to pass. Thus we return "impossible", which is correct.

Solution

Note that the proof of correctness depends on the pseudocode you chose to write. If you had written the following pseudocode, the proof may need to look significantly different to be correct.

Let $i = 1$.
while $i \leq n$ and $A[i] \neq k$ **do**
    Update $i = i + 1$.
if $i \leq n$ **then**
    return i
else
    return "impossible"

In fact, this pseudocode could be easier to reason about compared to the one in the solution above, despite not being what most programmers would write. This is because all the return statements are at the end, so they are not significantly affecting the control flow.

You are encouraged to think through how you would prove this version of the algorithm correct (not for credit, just practice). Feel free to ask anyone about this at office hours.

Solution

Find a sample video solution corresponding to the first solution above on Canvas. It demonstrates appropriate use of a whiteboard, personal notes, and style of presentation.