CSC236 Week 12: Correct, Before & After

Hisbaan Noorani

December 2 – December 8, 2021

Contents

fancyverb

1 Recursive Binary Search

We define correctness of a program in terms of it running, terminating, and fullfilling what it sought out to do. Purpose: find position where either x is, or should be inserted.

A: list, non-decreasing, comparable.

x: value to search for, must be comparable.

b: beginning index of search.

e: ending index of our search.

```
def recBinSearch(x, A, b, e):
    if b == e:              #
        if x <= A[b]:       #                 1. b ≤ p ≤ e + 1
            return b        # p return        2. b < p ⟹ A[p − 1] < x
        else:               #                 3. p < e + 1 ⟹ A[p] ≥ x
            return e + 1    #
    else:
        m = (b + e) // 2 # midpoint  # ⌊(b+e)/2⌋
        if x <= A[m]:
            return recBinSearch(x, A, b, m)      #
        else:                                    # ..
            return recBinSearch(x, A, m + 1, e)  #
```

```
          < x                        >= x
|---------------------|--------------------|
b                     p                    e
```

2 Conditions, pre- and post-

- x and elements of A are comparable.
- e and b are valid indices, $0 \leq b \leq e < |A|$.
- $A[b..e]$ is sorted non-decreasing.

`RecBinSearch(x, A, b, e)` terminates and also returns index p.

- $b \leq p \leq e + 1$
- $b < p \implies A[p-1] < x$
- $p \leq e \implies x \leq A[p]$

(except for boundaries, returns p so that $A[p-1] < x \leq A[p]$)

Prove that postcondition(s) follow from preconditions by induction on $n = e - b + 1$ (which is the size of the list).

3 Precondition $\implies$ termination and postcondition

Proof:

- Base Case, $n = 1$: Terminates because ther eare no loops or futher calls, returns $p = b = e \iff x \leq A[b = p]$ or $p = b + 1 = e + 1 \iff x > A[b = p - 1]$, so the postcondition is satisfied. Notice that the choice forces if-and-only-if.
- Induction step: Assume $n > 1$ and that the postcondition is satisfied for inputs of size $1 \leq k < n$ that satisfiy the precondition, and the `RecBinSearch(A, x, b e)` when $n = e - b + 1 > 1$. Since $b < e$ in this case, the check on line 2 fails, and line 8 executes.
- Exercise: $b \leq m < e$ in this case. there are two cases, according to whether $x \leq A[m]$ or $x > A[m]$.
  - Case 1: $x \leq A[m]$.
    * Show that IH applies to `RecurtiveBinarySearch(x, A, b, m)`.

      $$n = e - b + 1 > m - b + 1 \geq 1$$

    * Translate the postcondition to `RecurtiveBinarySearch(x, A, b, m)` These are now are IH:
      1. $b \leq p \leq m + 1$
      2. $b < p \implies A[p-1] < x$
      3. $p \leq m \implies A[p] \geq x$
    * Show that `RecurtiveBinarySearch(x, A, b, e)` satisfies postcondition

1. The first precondition:

$$\begin{aligned} b &\le m+1 && \text{(by the IH)} \\ &\le e+1 \end{aligned}$$

2. The second precondition:

$$b > p \implies A[p-1] < x \qquad \text{(by the IH)}$$

3. The third precondition:

$$p \le e \implies p \le m$$

   Since $p = m+1 \implies A[p-1] = A[m] \implies A[m] < x$, which is a contradiction, so $p \ne m+1$, so we must have $p \le m$.

- Case 2: $x > A[m]$
  * Show that IH applies to `RecBinarySearch(A, x , m + 1, e)`. We must show that

$$\begin{aligned} n &= e+b+1 \\ &> e-(m+1)+1 \\ &\dots \\ &\ge 1 \end{aligned}$$

  * Translate postcondition to `RecBinarySearch(x, A, m + 1, e)`. These are now our IH:
    1. $m+1 \le p \le e+1$
    2. $m+1 < p \implies A[p-1] < x$
    3. $p \le e \implies A[p] \ge x$
  * Show that `RecBinarySearch(x, A, b, e)`
    1. The first precondition:

$$\begin{aligned} &p \le e+1 && \text{(by the IH)} \\ &b \le m+1 \le p && \text{(since } b \le m \text{ by exercise)} \end{aligned}$$

    2. The second precondition:

$$b < p \implies p = m+1 \text{ or } p > m+1 \qquad \text{(by the IH)}$$

       In the case where $p > m+1$, we can say that $A[p-1] < x$ by IH #2.

       In the case where $p = m+1$, we can say that $A[p-1] = A[m] < x$ by the last else statement.

    3. The third precondition:

$$p \le e \implies A[p] \ge x$$

4 Correctbess by design

Draw bictures of before, during, and after

- Precondition: A sorted, comparable with x.
- Postcondition: $0 \leq b \leq n$ and $A[0:b] < x \leq A[p:b-1]$

4.1 "Derive" conditions from pictures

We need some notation for mutation.

- e_i will be e at the end of the i^{th} loop iteration.
- b_i will be b at the end of the i^{th} loop iteration.
- m_i will be m at the end of the i^{th} loop iteration.

Precondition: A is a sorted list comparable to x elements, $n = |A| > 0$. $0 = b \leq e = n - 1$

Postcondition: $0 \leq b \leq n$ and all($[j < xforjinA[0:b]]$) and all($[k \geq xforkinA[b:n]]$)

For all natural numbers i, define $P(i)$: At the end of hte loop iteration i (if it occurs), $0 \leq b_i \leq e_i + 1 \leq n$ and $b_i, e_i \mathbb{N}$. And, $all([j < xforjinA[0:b_i]])$ and $all([k \geq xforkinA[e_i+1:n]])$

Prove for all $i \in \mathbb{N}$, $P(i)$ using simple induction:

5 Prove termination

Associate a decreasing seqence in $\mathbb{N}$ with loop iterations. It helps to add claims to the loop invariant.

We are tempted to "prove" that "eventaully" $b_i > e_i$. DO NOT EVER DO THIS. A more successful approach is to divise an expression linked to loop iteration i that is (1) a natural number, and (2) strictly decreases with each loop iteration. A decreasing sequence of natiral numbers must, by definition, be finite by the property of well ordering.

A good candidate for such a sequence is the "distance" of A being searched, i.e. $e_i - b_i + 1$. We want to prove that this expression is a natural number and is strictly decreasing. The expression beinga natrual number follows directly from $P(i)$. The expression being strictly decreasing has two cases.

- Case $A[m] < x$: So, $e_{i+1} = e_i$ and $b_{i+1} = m_{i+1} + 1$. Then:

$$\begin{aligned}
e_{i+1} + 1 - b_{i+1} &= e_i + 1 - m_{i+1} - 1 \\
&= e_i + m_{i+1} \\
&< e_i + 1 - m_{i+1} \\
&\leq e_i + 1 - b_i
\end{aligned}$$

- Case $A[m] \geq x$: So, $e_{i+1} = m_{i+1} - 1$ and $b_{i+1} = b_i$

$$\begin{aligned}
e_{i+1} + 1 - b_{i+1} &= m_{i+1} - 1 + 1 - b_i \\
&= m_{i+1} - b_i \\
&< m_{i+1} + 1 - b_i \\
&\leq e_i + 1 + b_i
\end{aligned}$$

In both cases, we have a decreasing sequence of natural numbers corresponding to the loop iteration. ■