Algorithms CS315 **Quiz 1 Answer Key** **Spring 2018**

1. (20 points) Consider the **searching problem**:

Input: A sequence of n number $A = \langle a_1, a_2, \ldots, a_n \rangle$ and a value v.

Output: An index i such that $v = A[i]$ or the special value NIL if v does not appear in A.

(a) (5 points) Write the pseudocode for a linear search which scans through the sequence looking for v.

Answer

Solution 1

```
     Linear-search(A,v)
1       for i= 1 to A.length          c1 t1
2           if A[i] = v then          c2 t2
3               return i              c3 1

4       return NIL                    c4 1
```

Solution 2

```
    Linear-search(A,v)
1       r := NIL                      c0 1
2       for i= 1 to A.length          c1 t1
3           if A[i] = v then          c2 t2
4               r = i                 c3 1

5       return r                      c4 1
```

(b) (10 points) Use a loop invariant to prove your algorithm is correct.

Answer

Solution 1

Loop invariant: At the beginning of the loop on line 1 the subarray $A[1..i-1]$ does not contain the element v.

Initialization: The index variable i is set to 1. Before the loop is run the subarray $A[1, 0]$ is empty. Hence it vacuously does not contain the element v as it has no elements.

Maintenance: Suppose the subarray $A[1..i-1]$ does not contain the element v. Then on the ith iteration either $A[i] = v$ on line 2 in which case value i is returned on line 3 and the loop terminated or $A[i] \neq v$. In the first case the subarray $A[1..i-1]$ does no contain the element v. In the second case $A[i] \neq v$ so we have verified that the subarray $A[1..i]$ does not contain v. So at the next iteration with $i + 1$ the loop invariant holds.

Termination: If v is found then i is returned and the loop invariant holds as the loop is short circuited. If A does not contain v then line 3 never executes and we return

NIL on line 4 with $i = n + 1$ where $n = A.length$. The subarray $A[1..n + 1 - 1]$ is the complete array and it does not contain the element v. Hence our return value NIL is correct.

Solution 2

Loop invariant: At the beginning of the loop on line 2 the subarray $A[1..i-1]$ does not contain the element v until either v is found or the loop completes.

Initialization: The index variable i is set to 1. Before the loop is run the subarray $A[1, 0]$ is empty. Hence it vacuously does not contain the element v as it has no elements.

Maintenance: Suppose the subarray $A[1..i - 1]$ does not contain the element v. Then on the ith iteration either $A[i] = v$ on line 3 in which case value r is set to i on line 4 and the loop invariant holds. Otherwise $A[i] \neq v$ so we have verified that the subarray $A[1..i]$ does not contain v. So at the next iteration with $i+1$ the loop invariant holds.

Termination: On termination of the loop if r is not equal to `NIL` and $A[r] = v$ and in line 5 r is returned. If A does not contain v then on termination of the look. All $A[1..i + 1 - 1] = A$ elements are all not equal to v. Hence r remains set to `NIL` which is returned.

(c) (5 points) Give tight upper and lower asymptotic bounds for the runtime of your algorithm.

Answer

Solution 1

The times $t1$ and $t2$ can equal 1 if $A[1] = v$. In fact in this case $T(n) = c1+c2+c3$.

If v is not in A then $t1 = n + 1$ and $t2 = n$. And the run time $T(n) = c1(n + 1) + c2n + c4$.

Therefore $T(n) = \Omega(1)$ and $T(n) = O(n)$ are tight lower and upper asymptotic bounds for $T(n)$.

Solution 2

The times $t1$ and $t2$ are $n + 1$ and n respectively regardless whether v is found or not. Hence the runtime is $T(n) = C0 + C1(n+1) + C2(n) + C3 + C4$ if found and $T(n) = C0+C1(n+1)+C2(n)+C4$ if v is not found. Hence The tight upper and lower bounds for $T(n)$ is $O(n)$ and $\Omega(n)$ i.e. $T(n) = \Theta(n)$.