CIS 575. Introduction to Algorithm Analysis
Material for February 16, 2024

Correctness of Recursive Insertion Sort

1 Correctness Proof for Recursive Insertion Sort

Recall that in the first week we developed the recursive algorithm

```
InsertionSort(A[1..n])
  if n > 1
    InsertionSort(A[1..n − 1])
    InsertLast(A[1..n])
```

so as to implement the postcondition:

> $A[1..n]$ is a permutation of its original values such that $A[1..n]$ is non-decreasing.

The specification of InsertLast is given by

Precondition $A[1..\mathbf{n-1}]$ is non-decreasing

Postcondition $A[1..\mathbf{n}]$ is non-decreasing, and a permutation of its original values.

We shall prove the correctness of InsertionSort in two *independent* steps:

- prove that InsertionSort is correct *assuming* InsertLast meets its specification
- present an implementation of InsertLast that meets its specification.

1.1 Correctness Proof for InsertionSort

Assuming that InsertLast meets its specification, we shall prove by induction in n that for all $n \geq 0$, after calling InsertionSort on $A[1..n]$ it will hold that

> $A[1..n]$ is a permutation of its original values with $A[1..n]$ is non-decreasing.

We do a case analysis on the value of n:

- if $n \leq 1$, InsertionSort does nothing, which fulfills our requirements since an array with zero elements or one element is trivially non-decreasing.

- if $n > 1$, we have two function calls:

  1. InsertionSort($A[1..n-1]$) will transform A to some A'';
  2. InsertLast($A[1..n]$) will transform A'' to the result A'.

  Our goal is to prove that $A'[1..n]$ is a permutation of $A[1..n]$ with $A'[1..n]$ non-decreasing. Since $0 \leq n-1 < n$, we can apply the **induction hypothesis** to the call InsertionSort($A[1..n-1]$) to infer that

  - $A''[1..n-1]$ is a permutation of $A[1..n-1]$, and hence (due to our convention that functions do not access an array outside of the provided interval) also

  $$A''[1..n] \text{ is a permutation of } A[1..n] \tag{1}$$

  - $A''[1..n-1]$ is non-decreasing; this is the precondition for the call to InsertLast so from our assumption that InsertLast satisfies its specification we infer

  $$A'[1..n] \text{ is non-decreasing} \tag{2}$$

  and also that

  $$A'[1..n] \text{ is a permutation of } A''[1..n]. \tag{3}$$

  Combining (1) and (3) we see that $A'[1..n]$ is a permutation of $A[1..n]$, which together with (2) is the desired postcondition.

We have proved the correctness of the implementation of InsertionSort, but under the assumption that InsertLast is implemented to meet its specification. We shall now show that this is indeed possible.

1.2 Correctness Proof for InsertLast

Already in the first week of class we saw how to implement InsertLast:

```
InsertLast(A[1..n])
  if n > 1 and A[n] < A[n − 1]
    A[n] ↔ A[n − 1]
    InsertLast(A[1..n − 1])
```

We shall now prove, by induction in n, that for all $n \geq 1$ this satisfies the specification of InsertLast given by

Precondition $A[1..\mathbf{n-1}]$ is non-decreasing

Postcondition $A[1..\mathbf{n}]$ is non-decreasing, and a permutation of its original values.

For that purpose, we do a case analysis:

- if $n \leq 1$ then InsertLast does nothing, which fulfills our requirements since a one-element array is non-decreasing.

- if $n > 1$ and $A[n-1] \leq A[n]$ then InsertLast also does nothing, which fulfills our requirements since in that case, $A[1..n-1]$ being non-decreasing implies $A[1..n]$ being non-decreasing.

- Otherwise, we have $n > 1$ and $A[n] < A[n-1]$, in which case

  1. the swap $A[n] \leftrightarrow A[n-1]$ will transform A to some A'';
  2. the recursive call of InsertLast will transform A'' to the result A'.

  Since $1 \leq n-1 < n$, the induction hypothesis tells us that the recursive call of InsertLast meets its specification. Since $A[1..n-1]$ is non-decreasing, also $A[1..n-2] = A''[1..n-2]$ is non-decreasing. The precondition for the recursive call of InsertLast is thus satisfied, and we infer that the call will establish its postcondition:

  $$A'[1..n-1] \text{ is a permutation of } A''[1..n-1] \tag{4}$$

  and also

  $$A'[1..n-1] \text{ is non-decreasing.} \tag{5}$$

  Since the recursive call of InsertLast will not touch $A[n]$, we see from (4) that $A'[1..n]$ is a permutation of $A''[1..n]$ which obviously is a permutation of $A[1..n]$, and hence

  $A'[1..n]$ is a permutation of $A[1..n]$.

  We are left with showing that $A'[1..n]$ is non-decreasing, which by (5) can be done by showing that $A'[n-1] \leq A'[n]$. To see this, observe that there exists $i \in 1..n$ such that $A'[n-1] = A[i]$, and that $A'[n] = A''[n] = A[n-1]$. It thus suffices to show that $A[i] \leq A[n-1]$ for all $i \in 1..n$, which when $i \leq n-1$ follows from the precondition ($A[1..n-1]$ is non-decreasing), and when $i = n$ follows from our case assumption.

This concludes the proof that the implementation of InsertLast is correct.