Proving Insertion sort again

Mathematical induction

Prof Hans Georg Schaathun

Høgskolen i Ålesund

Autumn 2013 – Induction 4
Recorded: September 12, 2014

Insertion sort

Exercise

Prove that the output array of insertion sort (as given in previous videos) is sorted in increasing order.

```
procedure InsertionSort(A_1, A_2, ..., A_n)
  if n = 0, then return []
  else
    InsertionSort(A_1, A_2, ..., A_{n-1})
    insert(A_n into A_1, A_2, ..., A_{n-1})
```

Linking the problem to natural numbers

Different predicates compared to iterative case

$P(n) :=$ Insertion sort can correctly sort n numbers

- What is the base case?
- $P(0)$ the input is an empty array.
- $P(0)$ is trivially true

Linking the problem to natural numbers

Different predicates compared to iterative case

$$P(n) := \text{Insertion sort can correctly sort } n \text{ numbers}$$

- What is the base case?
- $P(0)$ the input is an empty array.
- $P(0)$ is trivially true

Linking the problem to natural numbers

Different predicates compared to iterative case

$$P(n) := \text{Insertion sort can correctly sort } n \text{ numbers}$$

- What is the base case?
- $P(0)$ the input is an empty array.
- $P(0)$ is trivially true

Inductive case

We need to prove that $P(n-1) \Rightarrow P(n)$.

- Two operations in the recursive case
  1. InsertionSort on $n-1$ numbers.
  2. Insert nth number into a sorted array.
- InsertionSort is correct for $n-1$ ($P(n-1)$)
- It suffices to prove that the insert subroutine is correct

The insert algorithm

```
procedure insert(e into A_1, A_2, ..., A_n)
  if n = 0, then
      A_1 = e
  else if e > A_n, then
      A_{n+1} = e
  else
      A_{n+1} = A_n
      insert(e into A_1, A_2, ..., A_{n-1})
```

- $Q(n)$: insert is correct for an array of n elements
- $Q(0)$: trivially true; a singleton is always sorted

Inductive case

- Two subcases
  1. $e > A_n$: e belongs at the end
  2. $e \leq A_n$: e is to the left of A_n
- In the second case, we insert into the subarray $A_1, A_2, \ldots, A_{n-1}$
- This we can do by the inductive hypothesis
- The insert algorithm is correct by induction.

Summarising

- $P(n)$: Insertion sort is correct for n inputs
- $P(0)$ is trivially true (empty array)
- $P(n-1) \Rightarrow P(n)$ because the insert operation is correct
- Hence $P(n)$ for any $n \geq 0$ by mathematical induction

Exercise

Selection sort

Consider the following selection sort algorithm (or the recursive variant which you have done in a previous exercise).

Input Array `A` of length `n`

Output The same array `A` sorted in place.

$$
\begin{array}{l}
\textbf{for } i := 1 \textbf{ to } n-1 \\
\quad \textbf{for } j := i+1 \textbf{ to } n \\
\quad\quad \textbf{if } A_i > A_j \\
\quad\quad\quad \text{swap } A_i \textbf{ with } A_j
\end{array}
$$

Question *How do we know that it correctly produces a sorted array?*