Cosc 201
Algorithms and Data Structures
Lecture 5 (10/3/2024)
Induction II

Brendan McCane
brendan.mccane@otago.ac.nz
Michael Albert

UNIVERSITY OF OTAGO
Te Whare Wānanga o Otāgo
NEW ZEALAND

Today's plan

- Traditional induction examples
- The link between induction and recursion

Sum of odd numbers

What is the sum of the first n odd numbers?

How does the preparation go?

It looks like the property that we want to prove is that the sum of the first n odd numbers is n^2.

Execution

- We will show that for any positive integer n the sum of the first n odd numbers, i.e.,
$$1 + 3 + 5 + \cdots + (2n - 1)$$
is n^2.
- For $n = 1$ the result is true, because the left hand side is 1, and the right hand side is 1^2 which is also equal to 1.
- For any $k \geqslant 1$, assuming the result is true for k, the result is true at $k + 1$ because of the following calculation:
$$\begin{aligned} 1 + 3 + 5 + \cdots + (2k - 1) + (2(k + 1) - 1) &= k^2 + 2k + 1 \\ &= (k + 1)^2. \end{aligned}$$
- Therefore, by induction, the result is true for all n.

Another recurrence

Consider the recurrence:

$$
\begin{aligned}
a_0 &= 0, \\
a_1 &= 1, \\
a_n &= 4a_{n-1} - 4a_{n-2} \quad \text{for } n \geqslant 2.
\end{aligned}
$$

Find (and prove correct) a formula for a_n in terms of n.

How does the preparation go?

It looks like the property that we want to prove is that $a_n = n \times 2^{n-1}$.

Execution

- We will show that for any non-negative integer n, $a_n = n \times 2^{n-1}$.
- For $n = 0$ and $n = 1$ the result is true by direct verification.
- For any $k \geqslant 1$, assuming the result is true for k, the result is true at $k + 1$ because of the following calculation:

$$\begin{aligned} a_{k+1} &= 4a_k - 4a_{k-1} \\ &= 4\,k\,2^{k-1} - 4\,(k-1)\,2^{k-2} \\ &= (8-4)\,k\,2^{k-2} + 4 \times 2^{k-2} \\ &= k\,2^k + 2^k \\ &= (k+1)\,2^k \\ &= (k+1) \times 2^{(k+1)-1}. \end{aligned}$$

- Therefore, by induction, the result is true for all n.

The link between induction and recursion

- In a recursive algorithm there are certain fundamental instances where the algorithm does not use recursion (the base cases).
- Other instances use recursion and apply the algorithm first to simpler instances, then combine the results in some way.
- The argument for correctness goes “well, if it works in the simpler cases, then it works in the more complicated ones because of the way we combine the results”.
- This is exactly the argument used in induction.
- In fact, we can usually isolate some parameter n associated with the algorithm so that, in the recursive calls, the parameter is always decreased. Then the claim that the algorithm works for all values of the parameter is ripe for a proof by induction.
- At a deep foundational level, induction and recursion are two sides of the same coin.