Introduction

I'm a Ph.D. student in the computer science department. My website, which will contain all this info and perhaps some notes, is `http://www.cs.pitt.edu/~mnugent/`. My office is Sennott Square 6505, and my office hours are Monday 11-1 and Wednesday 2-4, and my email is mpn1@pitt.edu. Feel free to set up an appointment if none of my regular office hours work for you.

Induction

Induction is used to prove an infinite number of statements at a time that can be somehow ordered. There is always a base case and the inductive step. In the base case, you prove directly that the first statement is true. In the inductive step, you assume that the nth statement is true and prove the $n+1$st statement is true.

Example 1

Now we'll prove $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Recall that

$$\sum_{i=1}^{n} i = 1 + 2 + ... + n - 1 + n$$

First, try it out with $n = 1, 2, 3....$ If I showed it worked for every number from 1 to 100 (or 1,000 or 1,000,000), would that prove that it's true? No. However, induction does, because it takes a general case and proves it's true.

Base Case: This is the case when $n = 1$. It's true because:

$$1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1$$

Inductive Step: Assume as the inductive hypothesis that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. We want to prove that $\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2}$. First observe that

$$\sum_{i=1}^{n+1} i = n + 1 + \sum_{i=1}^{n} i$$

which by the inductive hypothesis is

$$n + 1 + \frac{n(n+1)}{2}$$

And the rest is just algebra:

$$n+1+\frac{n(n+1)}{2} = \frac{2(n+1)}{2} + \frac{n(n+1)}{2}$$
$$= \frac{2(n+1)+n(n+1)}{2} = \frac{(n+2)(n+1)}{2}$$
$$= \frac{(n+1)((n+1)+1)}{2}$$

And we're done. This shows that (for example) this is true for $n = 1000$, since it proves that it's true as long as it's true for $n = 999$, which we know is true as long as it's true for $n = 998$, etc, until we get to $n = 1$, which we proved directly.

Example 2

Next we prove by induction that the number of leaves in a complete binary tree of depth d is 2^d. Recall that a complete binary tree is one where all non-leaf nodes have two children, and each leaf is d nodes away from the root.

Base Case: This is when a tree has depth 0. In this case, the only node there is the root node. That node has no children, so it is a leaf, so the number of leaves is 2^0.

Inductive Step: Assume as the inductive hypothesis that a tree of depth d has 2^d leaves. For a tree of depth $d+1$, note that it can be created by taking a tree of depth d and adding two children to every leaf. Thus the number of children in a tree of depth $d+1$ is $2 \cdot 2^d = 2^{d+1}$.

Example 3

Now we prove the number of internal nodes of a complete binary tree of depth d is $2^d - 1$.

Base Case: Again, this is when a tree has depth 0, so the only node is the root, and it is a leaf, not an internal node, so there are 0 internal nodes, and $2^0 - 1 = 1 - 1 = 0$.

Inductive Step: We assume that a tree of depth d has $2^d - 1$ internal nodes. For a tree of depth $d+1$, note that all of the nodes that used to be leafs become internal, and there were 2^d leaf nodes, so a depth $d+1$ tree has $2^d - 1 + 2^d$ internal nodes, and this is the same as $2 \cdot 2^d - 1 = 2^{d+1} - 1$ internal nodes.

Example 4

Now we prove that $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$

Base Case: $1^2 = 1$, and $\frac{1(1+1)(2+1)}{6} = \frac{6}{6} = 1$.

Inductive Step: Assume the statement is true for n. Then for $n+1$ we have

$$\sum_{i=1}^{n+1} i^2 = (n+1)^2 + \sum_{i=1}^{n} i^2 = (n+1)^2 + \frac{n(n+1)(2n+1)}{6} = \frac{6(n+1)^2 + n(n+1)(2n+1)}{6}$$
$$= \frac{(n+1)(6(n+1) + n(2n+1))}{6} = \frac{(n+1)(6n+6+2n^2+n)}{6} = \frac{(n+1)(2n^2+7n+6)}{6}$$
$$= \frac{(n+1)(n+2)(2n+3)}{6} = \frac{(n+1)((n+1)+1)(2(n+1)+1)}{6}$$