Mathematical Induction

Mathematical induction is usually presented as a method for establishing identities like

$$1^2 + 2^2 + \dots n^2 = \frac{n(n+1)(2n+1)}{6}$$

And the steps are described along these lines:

1. Show that the identity is true for $n = 1$,
2. assume that it's true for n, and
3. use this assumption to prove that it's true for $n + 1$.

We've found that this is baffling to many students. If you present the induction process in this way, you'll often hear two dreaded questions from your class.

Dreaded question 1: *Where did the identity come from in the first place?*

Dreaded question 2: *How can I assume that it's true for n—isn't that what we're supposed to* ***prove****?*

After several years of heartburn teaching induction, we came up with a different approach that actually seems to make sense to students. It uses technology in a basic way. In the early days when we were developing this method, we used programming languages like Logo or Scheme. The language on the TI-89 allowed us to implement the method on handhelds, but it involved some cumbersone computerish details. But the "function modeling language" on the TI-Nspire^{tm} is a perfect medium for our method.

One indication that it makes sense is that the dreaded questions don't come up in class.

In this session, we'll look at how to use the Nspire to help students understand mathematical induction.

Here we go. . .

 Adapted from *The CME Project: Precalculus* forthcoming from Pearson, 2008

1. For each problem, find a function that fits the table.
Be clever.

(a)

INPUT	OUTPUT
0	1
1	3
2	5
3	7
4	9

(b)

INPUT	OUTPUT
0	-7
1	-4
2	-1
3	2
4	5

Habits: Model your functions in your FML. Find different models that fit the table.

(c)

INPUT	OUTPUT
0	1
1	2
2	5
3	10
4	17

(d)

INPUT	OUTPUT
0	0
1	1
2	3
3	6
4	10

These problems are not only fun—they also get you in shape for the Monthly Payments workshop with Jean Benson and Doreen Kilday.

(e)

INPUT	OUTPUT
0	2
1	3
2	5
3	9
4	17

(f)

INPUT	OUTPUT
0	0
1	1
2	4
3	9
4	16

(g)

INPUT	OUTPUT
0	0
1	1
2	4
3	10
4	20

(h)

INPUT	OUTPUT
0	1
1	$\frac{1}{2}$
2	$\frac{1}{3}$
3	$\frac{1}{4}$
4	$\frac{1}{5}$

Stop. Let's take a look at a closed form definition and a recursive definition for functions that fit table 1.a. Are the functions equal? Then let's do the same for table 1.d.

What does it mean for two functions to be *equal*?

Tabulate each function below from 0 to 5, then find a closed-form definition for a function that agrees with the table.

2.

$$h(n) = \begin{cases} 3 & \text{if } n = 0 \\ h(n-1) + 8 & \text{if } n > 0 \end{cases}$$

3.

$$f(m) = \begin{cases} 0 & \text{if } m = 0 \\ f(m-1) + 2m & \text{if } m > 0 \end{cases}$$

4.

$$c(m) = \begin{cases} 3 & \text{if } m = 0 \\ c(m-1) + m & \text{if } m > 0 \end{cases}$$

5.

$$j(t) = \begin{cases} -1 & \text{if } t = 0 \\ j(t-1) + 2t & \text{if } t > 0 \end{cases}$$

6. Find a recursive definition for a function that fits this table.

n	$T(n)$
0	1
1	3
2	9
3	27
4	81
5	243

7. Find a recursively-defined function that fits this table.

x	$Z(x)$
0	3
1	10
2	21
3	36
4	55

8. Find a recursively-defined function that fits this table.

n	$\Gamma(n)$
0	1
1	1
2	2
3	6
4	24
5	120

 Adapted from *The CME Project: Precalculus* forthcoming from Pearson, 2008