5. Induction

Discrete Mathematics

Mathematical induction

Mathematical induction is a method of *mathematical proof*...

...often used to prove that something is true for all natural numbers

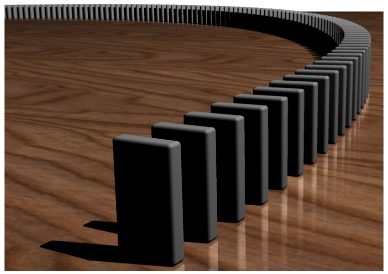
Method

Mathematical induction has two parts:

1. Prove that it is true for one (*base case*)

2. Prove that if it is true for n then it is also true for n+1 (*inductive step*).

Dominos



Base case:

The first domino will fall.

Inductive step:

If a domino falls, then the next domino will also fall.

→ All the dominos will fall!

Example proofs

Prove that the following statement is true for all natural numbers.

$$0 + 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

Base case (n=0):

$$0 = \frac{0 \cdot (0+1)}{2}.$$

Inductive step:

$$(0 + 1 + 2 + \cdots + n) + (n+1) = \frac{(n+1)((n+1)+1)}{2}$$

because...

$$\begin{aligned} \frac{n(n+1)}{2} + (n+1) &= \frac{n(n+1) + 2(n+1)}{2} \\ &= \frac{(n+1)(n+2)}{2} \\ &= \frac{(n+1)((n+1)+1)}{2}. \end{aligned}$$

Example proofs

Question. Use mathematical induction to prove that $2^0 + 2^1 + 2^2 + \ldots + 2^n = 2^{n+1} - 1$

Answer.

Base case (n=0): $2^0 = 1 = 2^{0+1} - 1$

Inductive step:

$$\begin{aligned} (2^0 + 2^1 + 2^2 + \ldots + 2^n) + 2^{n+1} &= (2^{n+1} - 1) + 2^{n+1} \\ &= 2 * 2^{n+1} - 1 \\ &= 2^{n+2} - 1 \\ &= 2^{(n+1)+1} - 1 \end{aligned}$$

Example proofs

Question. Show that $n < 2^n$ for all positive integers n.

Answer.

Base case (n=1): $1 < 2^1$

Inductive step:

$$\begin{aligned} & n < 2^n \\ & n + 1 < 2^n + 1 && \text{(add one to both sides)} \\ & n + 1 < 2^n + 2^n && \text{(because } 1 \le 2^k\text{)} \\ & n + 1 < 2 * 2^n \\ & n + 1 < 2^{n+1} \end{aligned}$$

Example proofs

Question. Prove that $n^3 - n$ is divisible by 3 for all positive integers n.

Answer.

Base case (n=1): $1^3 - 1 = 0$ (=> is divisible by 3)

Inductive step:

$$\begin{aligned} (n+1)^3 - (n+1) &= (n^3 + 3n^2 + 3n + 1) - (n+1) \\ &= n^3 - n + 3n^2 + 3n \\ &= (n^3 - n) + 3(n^2 + n) \end{aligned}$$

Divisible by 3   Divisible by 3

Principle of mathematical induction

To prove that P(n) is true for all natural numbers, where P(n) is a propositional function, we must complete two steps:

Base case

Show that P(0) is true (or that P(1) is true for positive integers).

Inductive step

Show that if P(n) is true then P(n+1) is also true (where n is a natural number).

Applications in computer science

One example...

Recursive algorithms are closely related to mathematical induction.

(Sometimes called "divide and conquer" algorithms)

Recursive algorithm example 1

Factorial(5) = 5 * 4 * 3 * 2 * 1 = 5!

```
int Factorial(int n) {
    if (n == 0)
        return 1;                  // Base case: Fac(0) = 1
    else
        return n * Factorial(n-1); // Inductive step: Fac(n+1) = (n+1)*Fac(n)
}
```


Recursive algorithm example 2

Power(2,4) = 2^4 = 2 * 2 * 2 * 2

Write a recursive algorithm for Power(a,b), e.g.

```
int Power(int a, int b) {
    ?
}
```

Recursive algorithm example 2

Power(2,4) = 2^4 = 2 * 2 * 2 * 2

```
int Power(int a, int b) {
    if (b == 0)
        return 1;              // Base case: Power(a,0) = 1
    else
        return a * Power(a, b-1);   // Inductive step: Power(a,b+1) = a*Power(a,b)
}
```


Recursive algorithm example 3

You have an array of n numbers, e.g. {1, 5, 6, 2, 4}

Write a recursive function that returns true if the array contains the number x.

(This is called a recursive linear search!)

```
bool Contains(int[] array, int x) {
    if (array.Count() == 0)    // Base case: Contains({},x) = false
        return false;
    else            // Inductive step: Contains({a,...},x) = (a == x) or Contains({...},x)
        return array[0] == x || Contains(array.RemoveFirst(), x);
}
```


Vocabulary

Words you must remember!

Mathematical induction (or "proof by induction")

Proof (noun from "to prove")

Base case

Inductive step

For all natural numbers (means 0,1,2,3,4,...)

For all positive integers (means 1,2,3,4,...)

Recursive algorithm (or "recursive function")

The library contains books!

"Discrete Mathematics and its applications" by Rosen

Read Chapter 4 (Sections 4.1 Induction & 4.4 Recursion)