Matrices

Are grids of numbers…they are really, really useful to computer scientists…at the end of the day, an image in its raw representation, is usually just a grid of numbers, where those numbers specify colors.

A matrix has some number rows and columns. Here is a 2 by 3 matrix:

$$\begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 9 \end{bmatrix}$$

Add Matrices

Dimensions have to be identical:

$$\begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 9 \end{bmatrix} + \begin{bmatrix} 6 & 8 & 2 \\ -5 & 1 & 7 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 5 \\ -1 & 3 & 16 \end{bmatrix}$$

Just add corresponding terms to get the resultant term

Subtraction is the same subtract first term minus second term for each slot:

$$\begin{bmatrix} 2 & -1 & 3 \\ 4 & 2 & 9 \end{bmatrix} - \begin{bmatrix} 6 & 8 & 2 \\ -5 & 1 & 7 \end{bmatrix} = \begin{bmatrix} -4 & -9 & 1 \\ 9 & 1 & 2 \end{bmatrix}$$

Multiplication of Matrices

Let M1 have r1 rows, c1 cols

Let M2 have r2 rows, c2 cols

We can multiply M1 x M2, if and only if c1 = r2. (Note: it's possible that we can multiply M1 times M2, but NOT be allowed to multiply M2 times M1.)

The result of the multiplication is always has r1 rows and c2 columns:

$$\begin{bmatrix} 1 & 2 & 3 \\ -2 & 4 & -1 \end{bmatrix} \times \begin{bmatrix} -3 & 5 \\ -4 & 6 \\ 0 & -5 \end{bmatrix} = \begin{bmatrix} 1(-3)+2(-4)+3(0) & 1(5)+2(6)+3(-5) \\ -2(-3)+4(-4)+(-1)(0) & -2(5)+4(6)+(-1)(-5) \end{bmatrix}$$

$$= \begin{bmatrix} -11 & 2 \\ -10 & 19 \end{bmatrix}$$

In general, to get the entry in row x, column y, we "multiply" row x by column y. What it means to multiply a row by a column, is roughly the definition of a dot product, if you happen to be familiar with that concept. If you are not, what we do is multiply each corresponding term and add those values.

$$\text{Result[x][y]} = \sum_{i=1}^{c1} M1[x][i] * M2[i][y]$$

Matrix Multiplication in code:

```
public static int[][] mult(int[][] m1, int[][] m2) {
    int[][] res = new int[m1.length][m2[0].length];
    for (int i=0; i<m1.length; i++)
       for (int j=0; j<m2[0].length; j++)
           for (int k=0; k<m1[0].length; k++)
               res[i][j] += (m1[i][k]*m2[k][j]);
    return res;
}
```

Recursively Defined Sequences
Fibonacci – wondered…

Month 1 – 1 pair of rabbits (new)
Month 2 – 1 pair of rabbits (mature)
Month 3 – 2 pairs of rabbits (1 mature, 1 new)
Month 4 – 3 pairs of rabbits (2 mature, 1 new)
Month 5 – 5 pairs of rabbies (3 mature, 2 new)

Not too hard to see…all rabbits in month k-1 become mature in month k. All rabbits alive in month k-2 give birth to new rabbits in month k.

Let F_k = # of pairs of rabbits alive at month k.
Then for k >= 3, by the definition of the story, we have

$F_1 = 1$, $F_2 = 1$, for all k > 2, we have $F_k = F_{k-1} + F_{k-2}$

There is a magazine called the Fibonacci Quarterly…if you can believe that…

Hallmarks of a recursively defined sequence:

Some initial terms are given.
To generate the rest of the terms, a formula is given to find the kth term in terms of previous terms of the sequence.

Here is a recursive definition of factorials:

0! = 1
for all k > 0, k! = k * (k-1)!

Here is a recursive definition of the combination function:

C(n, 0) = C(n, n) = 1, for all non-negative integers n.
C(n, k) = C(n-1, k-1) + C(n-1, k) for all n > 1 and 0 < k < n.

The most simple way to list a particular term in a recursively defined sequence is to build it up…

What is F_{10}?
1,1,2,3,5,8,13,21,34,55

Answer is 55…

Mathematical Induction
f(n) is an open statement where n can be any non-negative integer.

Let's pretend we proved the following two things:

f(0) and
$f(0) \rightarrow f(1)$

Using the rules of inference, what could we deduce?

f(0)	Premise
$f(0) \rightarrow f(1)$	Premise
f(1)	Modus Ponens

Now, imagine that the implication that we proved wasn't just about f(0), but about f(k) for all non-negative integers k:

Given
----
f(0)
$f(k) \rightarrow f(k+1)$, for all non-negative integers k…expand out a bit…
$f(0) \rightarrow f(1)$
$f(1) \rightarrow f(2)$
$f(2) \rightarrow f(3)$
$f(3) \rightarrow f(4)$, … on forever

Proof
----

f(0)	Premise
$f(0) \rightarrow f(1)$	Premise
f(1)	Modus Ponens
$f(1) \rightarrow f(2)$	Premise
f(2)	Modus Ponens
$f(2) \rightarrow f(3)$	Premise
f(3)	Modus Ponens
$f(3) \rightarrow f(4)$	Premise

…
This will go forever, and it will prove the statement for any arbitrary positive integer…

So, to prove any arbitrary statement, f(n), true for all non-negative integers n, it suffices to prove the two following things:

f(0)
f(k) → f(k+1) for all non-negative integers k

More generally, to prove any arbitrary statement, f(n), true for all integers n >= c, for some fixed integer c, it suffices to prove the two following things:

f(c)
f(k) → f(k+1) for all integers k >= c.

Normally, we break this up into three steps in mathematical induction:

1. Base case – this is where we prove f(c)
2. Inductive Hypothesis: This is were we assume f(k) for an arbitrary integers n=k, where k>=c.
3. Inductive Step: We must prove f(k+1).

Essentially, we are using direct proof to prove the if-then statement which is the second requirement I had previously written down.

Example #1

Prove for all positive integers n, that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

Base case: n = 1, LHS = $\sum_{i=1}^{1} i = 1$, RHS = $\frac{1(1+1)}{2} = \frac{2}{2} = 1$, the statement is true for n=1, thus the base case is proven.

Inductive hypothesis: Assume for an arbitrarily chosen positive integer n = k that$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$.

Inductive Step: Prove for n=k+1 that $\sum_{i=1}^{k+1} i = \frac{(k+1)((k+1)+1)}{2} = \frac{(k+1)(k+2)}{2}$

Quick Side Note: When proving A = B, **NEVER** manipulate both sides of the equation. Instead, start with one expression, and do algebra on it until you get to the other:

A = A1
= A2

= A 3
= …
= B

1+2+3+…+k+(k+1) = [1 + 2 + 3 +…+k] + (k+1)

$\sum_{i=1}^{k+1} i = \left(\sum_{i=1}^{k} i\right) + (k+1)$, note, I just split off the last term of the sum

$= \frac{k(k+1)}{2} + (k+1)$, using the inductive hypothesis (IH)

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

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

This proves the inductive step. It follows that for all positive integers n, $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$, as desired.

Many teachers just use induction to prove summation formulas, but it's really important for me to prove to you that induction is a general proof technique that can be used to prove assertions about positive integers in almost any mathematical realm!!!

Divisibility Proof

Prove for all non-negative integer n that $5 \mid (3^{2n} + 4^{n+1})$.

Base case: n = 0, plug in 0 for n in the expression: $3^{2(0)} + 4^{0+1} = 1 + 4 = 5$. Since 5 = 5(1), it is true that 5 | 5, thus the base case is true.

Inductive hypothesis: Assume for an arbitrarily chosen integer n = k that $5 \mid (3^{2k} + 4^{k+1})$. By definition of divisibility this means there exists some integer c such that $3^{2k} + 4^{k+1} = 5c$.

Inductive step: Prove that for n = k+1, that $5 \mid (3^{2(k+1)} + 4^{(k+1)+1})$.

$$\begin{aligned}
3^{2(k+1)} + 4^{(k+1)+1} &= 3^{2k+2} + 4^{k+2} \\
&= 3^{2k}3^{2} + 4^{k+1}4^{1} \\
&= 9(3^{2k}) + 4(4^{k+1}) \\
&= 5(3^{2k}) + 4(3^{2k}) + 4(4^{k+1}) \\
&= 5(3^{2k}) + 4[3^{2k} + 4^{k+1}] \\
&= 5(3^{2k}) + 4(5c) \\
&= 5(3^{2k} + 4c)
\end{aligned}$$

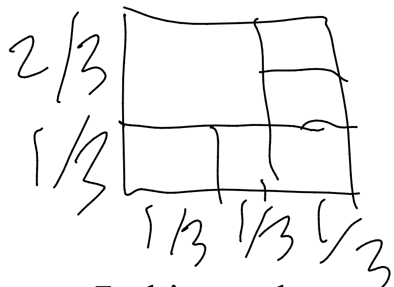
Since k is a non-neg integer and c is an integer, it follows that $(3^{2k} + 4c)$ is an integer. Thus, we can conclude as desired, that $5 \mid (3^{2(k+1)} + 4^{(k+1)+1})$

Induction Proof - n squares

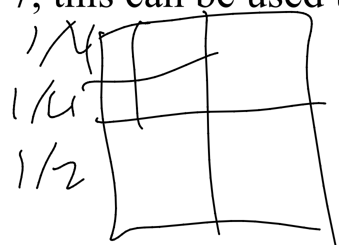
Thursday, October 8, 2020 2:44 PM

Use induction to prove that we can partition any square into n squares for all n >= 6.

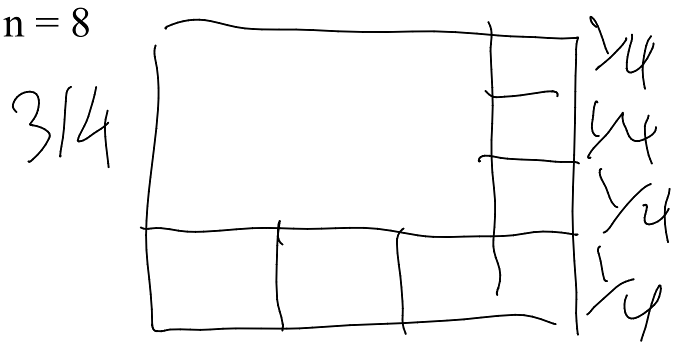
Base case n = 6…this can be used to build 6, 9,12,…



Base case n = 7, this can be used to build 7,10,13,…



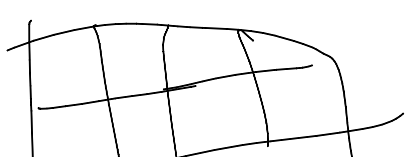
This one can be used to build 8,11,14, …
Base case n = 8

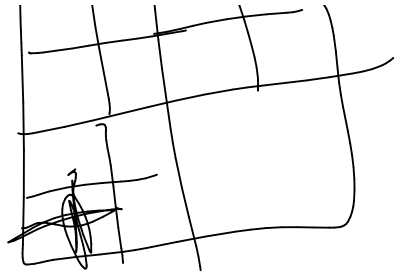


Inductive hypothesis: Assume for n=k-2, n=k-1 and n=k that we can partition a square into n squares, where k-2>=6, so k>=8.

Inductive step: Prove for n=k+1 that we can partition a square into n squares.

Take the IH, it gives you some arbitrary partition of a square into k-2 squares:





How can I use this picture to partition it into k+1 squares?

Answer: Take any square in the picture and split into 4 squares with half the side length of the square being split.

And we're done! We have proven that we can partition a square into k+1 squares!

Why did I need multiple base cases? I didn't build off the result of k, but I built off the result of k-2, so I needed three cases in a row.