

# Intro Math Problem Solving

## September 26

The Collatz Sequence

Two-step Stepping Stones

The Fibonacci Sequence

Thursday's Exam

# In Class Exercise

if  $n$  is even

divide it by 2

but if  $n$  is odd replace it by :

A)  $3n+1$

B)  $3n-1$

C)  $5n+1$

and continue until you reach 1.

We will talk about option A.

What can happen with options B or C?

# Recall "Stepping stone" Sequences

Instead of a formula for  $a(n)$ , some sequences start at  $a(0)$  or  $a(1)$ , then take one step at a time towards  $a(n)$ .

Computing even a single value  $a(n)$  nonetheless requires a FOR loop.



# Example: The Triangular Sequence

$a(i)$  counts the dots necessary to draw a triangle of dots with  $i$  rows.  $a(6) = 21$

```
  O
 OO
OOO
OOOO
OOOOO
OOOOOO
```

Formula:  $a(i) = i*(i+1)/2$

Stepping stone:

$$a(0) = 0$$

$$a(i) = a(i-1) + i$$

$$a(0) = 0$$

$$a(1) = a(0) + 1 = 0 + 1 = 1$$

$$a(2) = a(1) + 2 = 1 + 2 = 3$$

$$a(3) = a(2) + 3 = 3 + 3 = 6$$

$$a(4) = a(3) + 4 = 6 + 4 = 10$$

$$a(5) = a(4) + 5 = 10 + 5 = 15$$

$$a(6) = a(5) + 6 = 15 + 6 = 21$$

Insight Through

# triangular\_keep1.m

```
n = input ( 'Enter the highest index to compute: ' );
```

```
for i = 0 : n
```

```
    if ( i == 0 )
```

```
        a = 0;
```

```
    else
```

```
        aold = a;      ← We save a copy of a as "aold";
```

```
        a = aold + i;  ← and then we "step" to the next value.
```

```
    end
```

```
    fprintf ( 'a(%d) = %d\n', i, a );
```

```
end
```

# Why Stepping Stones?

So far, the sequences we have seen could be described equally well by a formula or by a stepping stone rule.

Today, we will see two examples for which a stepping stone rule is natural, but for which a formula does not exist, or is not at all obvious to you.

## Case 1: The Collatz Sequence

## A mysterious sequence

In weather, a hailstone forms from a particle of ice that grows through repeated rising and falling through cold rain clouds, until it grows so heavy it must fall to the ground.

Here's a mathematical calculation where we compute a sequence of numbers that can rise and fall until suddenly stopping at 1.

No one has a simple formula to predict how many such steps will be involved.

# The rules for "hailstone"

Pick an initial value for  $N$ .

If  $N$  is 1, we are done.

If  $N$  is even, replace  $N$  by  $N/2$ .

If  $N$  is odd, replace  $N$  by  $3*N+1$ .

Repeat.



# Hailstone Calculation for N = 12

0 12

1  $12/2=6$

2  $6/2=3$

3  $3*3+1=10$

4  $10/2 = 5$

5  $3*5+1 = 16$

6  $16/2 = 8$

7  $8/2 = 4$

8  $4/2 = 2$

9  $2/1 = 1$  stop after 9 steps

# hail\_next.m

Assuming  $I = 0$ , and  $N$  is defined, this will work, showing one value at a time.:

```
while ( true )
```

```
    fprintf ( ' %d: N = %d\n', i, n );
```

```
    pause
```

← Wait for me to hit RETURN

```
    if ( n == 1 )
```

← "If N = 1, stop"

```
        fprintf ( ' Terminate.\n' );
```

```
        i = 0;
```

```
        break;
```

```
    end
```

```
    i = i + 1;
```

```
    if ( mod ( n, 2 ) == 0 )
```

← Even numbers get divided by 2;

```
        n = n / 2;
```

```
    else
```

```
        n = 3 * n + 1;
```

← Odd numbers, triple plus 1.

```
    end
```

```
end
```

Insight Through

# Hailstone is a Stepping Stone Sequence

Symbolize the  $i$ -th entry as  $h(i)$ .

We get to pick  $h(0)$ , the initial value.

Then we apply the rule to get the next entry.

We can't jump ahead; to find  $h(10)$ , we have to compute  $h(1)$ ,  $h(2)$ , ...,  $h(9)$  first.

The sequence rule is more complicated than usual.

Also, it's unusual that the sequence stops (is finite, not infinite). [That's a conjecture.]

# Strategy for a program

Let the user pick an initial value for N

Let I count the steps, starting at 0.

AS LONG AS ( N is not 1 )

Take another step (increase I)

If ( N is even ) then

Divide it by 2

Otherwise ( it's odd ) so

Multiply it by 3 and add 1

Print N (it has changed!)

End of repeated statements

# hail\_sequence.m

```
i = 0;

while ( n ~= 1 )

    if ( i == 0 )
        n = input ( 'Enter starting value for Collatz sequence: ' );
    else
        if ( mod ( n, 2 ) == 0 )
            n = n / 2;
        else
            n = 3 * n + 1;
        end
    end
    fprintf ( ' %d', n );
    i = i + 1;

end

fprintf ( '\n' );
fprintf ( ' We took %d steps\n', i );
```

Insight Through

## This is a famous problem

This **hailstone** problem has other names. Sometimes it's called the  **$3n+1$**  problem. It was first described by mathematician Lothar Collatz, so we hear of the **Collatz problem**, or a **Collatz sequence**. The problem interests mathematicians because a simple set of rules reveal a combination of chaos and patterns.

# Chaos and Patterns

For a number  $N$ , let "length( $N$ )" count the steps until  $N$  is transformed to 1.

One pattern:  $\text{length}(2^k) = k$ .

So,  $\text{length}(64) = 6$ : 64, 32, 16, 8, 4, 2, 1.

"Chaos": length varies unpredictably:

$\text{length}(26) = 10$ , but  $\text{length}(27) = 111$

and no formula for sequence entries.

# hail\_length.m

```
i = 0;

while ( 1 < n )

    if ( i == 0 )
        n = input ( ' Enter a starting value for Collatz sequence: ' );
        n_save = n;
    else
        if ( mod ( n, 2 ) == 0 )
            n = n / 2;
        else
            n = 3 * n + 1;
        end
    end

    i = i + 1;

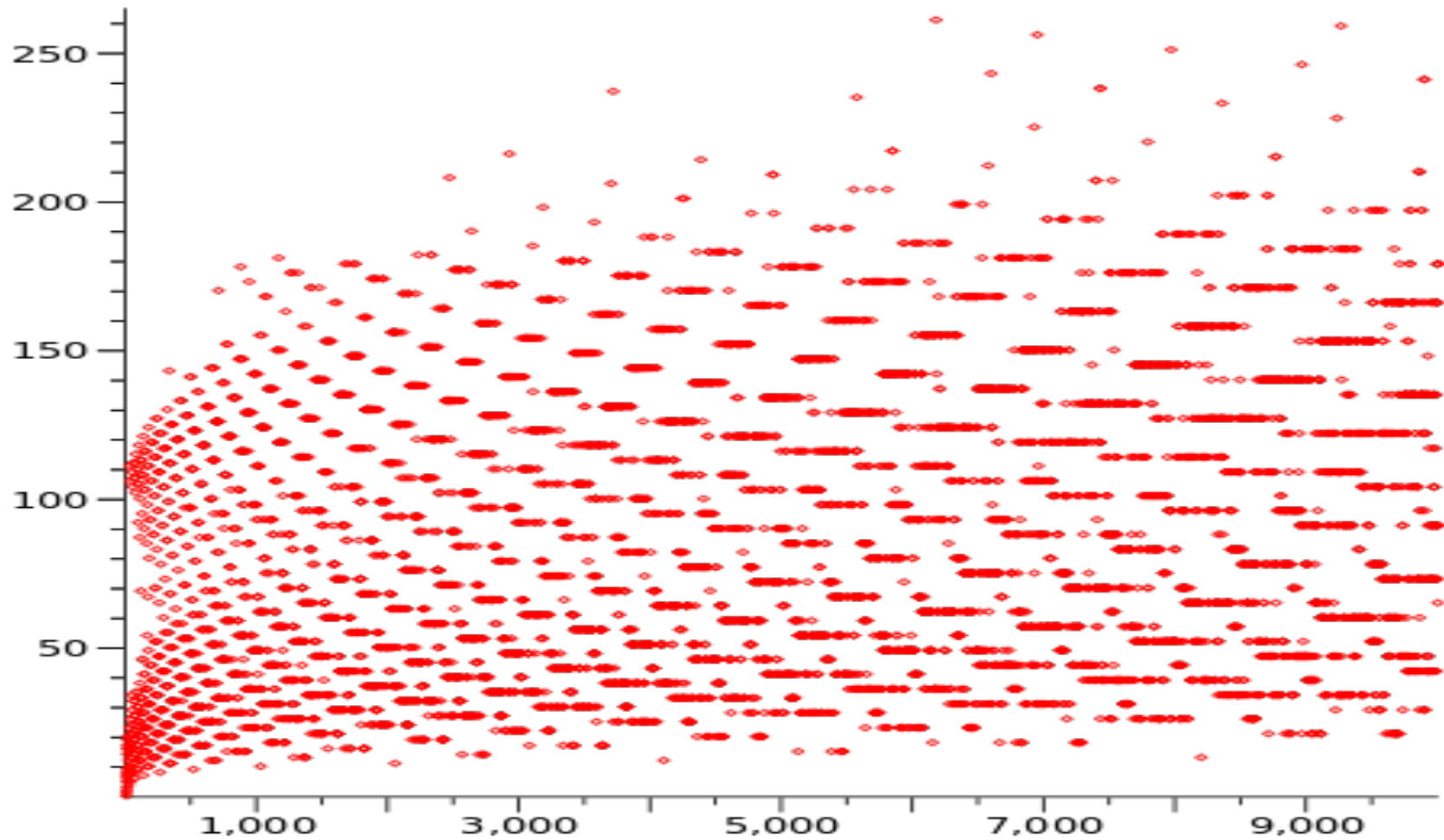
end

fprintf ( ' Collatz sequence for %d has length %d\n', n_save, i );
```

Insight Through



# The "length" for $1 \leq N \leq 10,000$



# The "Height" of a Collatz Sequence

Assuming every Collatz sequence

"hailstone" lands at 1 eventually, it's also interesting to ask how high it got during its up and down cycles.

To do this, we have to compute the maximum value of the sequence, which we do by creating a new variable "n\_max", and then comparing it to the value of "n" every time.

# collatz\_height.m

```
i = 0;

while ( 0 < n; )

    if ( i == 0 )
        n = input ( ' Enter a starting value for Collatz sequence: ' );
        n_save = n;
        height=n;
    else
        if ( mod ( n, 2 ) == 0 )
            n = n / 2;
        else
            n = 3 * n + 1;
        end
        height = max ( height, n );
    end

    i = i + 1;

end
fprintf ( ' Collatz sequence for %d has height %d\n', n_save, height );
```

Insight Through

# Collatz Sequence for N=27

**27**, 82, **41**, 124, 62, **31**, 94, **47**, 142, **71**, 214, **107**, 322, **161**, 484, 242, **121**, 364, 182, **91**, 274, **137**, 412, 206, **103**, 310, **155**, 466, **233**, 700, 350, **175**, 526, **263**, 790, **395**, 1186, **593**, 1780, 890, **445**, 1336, 668, 334, **167**, 502, **251**, 754, **377**, 1132, 566, **283**, 850, **425**, 1276, 638, **319**, 958, **479**, 1438, **719**, 2158, **1079**, 3238, **1619**, 4858, **2429**, 7288, 3644, 1822, **911**, 2734, **1367**, 4102, **2051**, 6154, **3077**, **9232**, 4616, 2308, 1154, **577**, 1732, 866, **433**, 1300, 650, **325**, 976, 488, 244, 122, **61**, 184, 92, 46, **23**, 70, **35**, 106, **53**, 160, 80, 40, 20, 10, **5**, 16, 8, 4, 2, **1**

Here is the Collatz sequence that starts at N=27.

The Collatz length is 111.

The Collatz height is 9,232.

Odd numbers (in **bold** ) make the sequence rise.

Who could predict this complicated behavior?

Insight Through

# Mathematical Questions

Does every starting value  $N$  eventually reach 1?

Does any starting value result in a loop?

Does any starting value produce an infinite, nonrepeating sequence?

Is there any way to predict starting values  $N$  that produce a sequence that takes a long time to reach 1, or that hits a very high value?

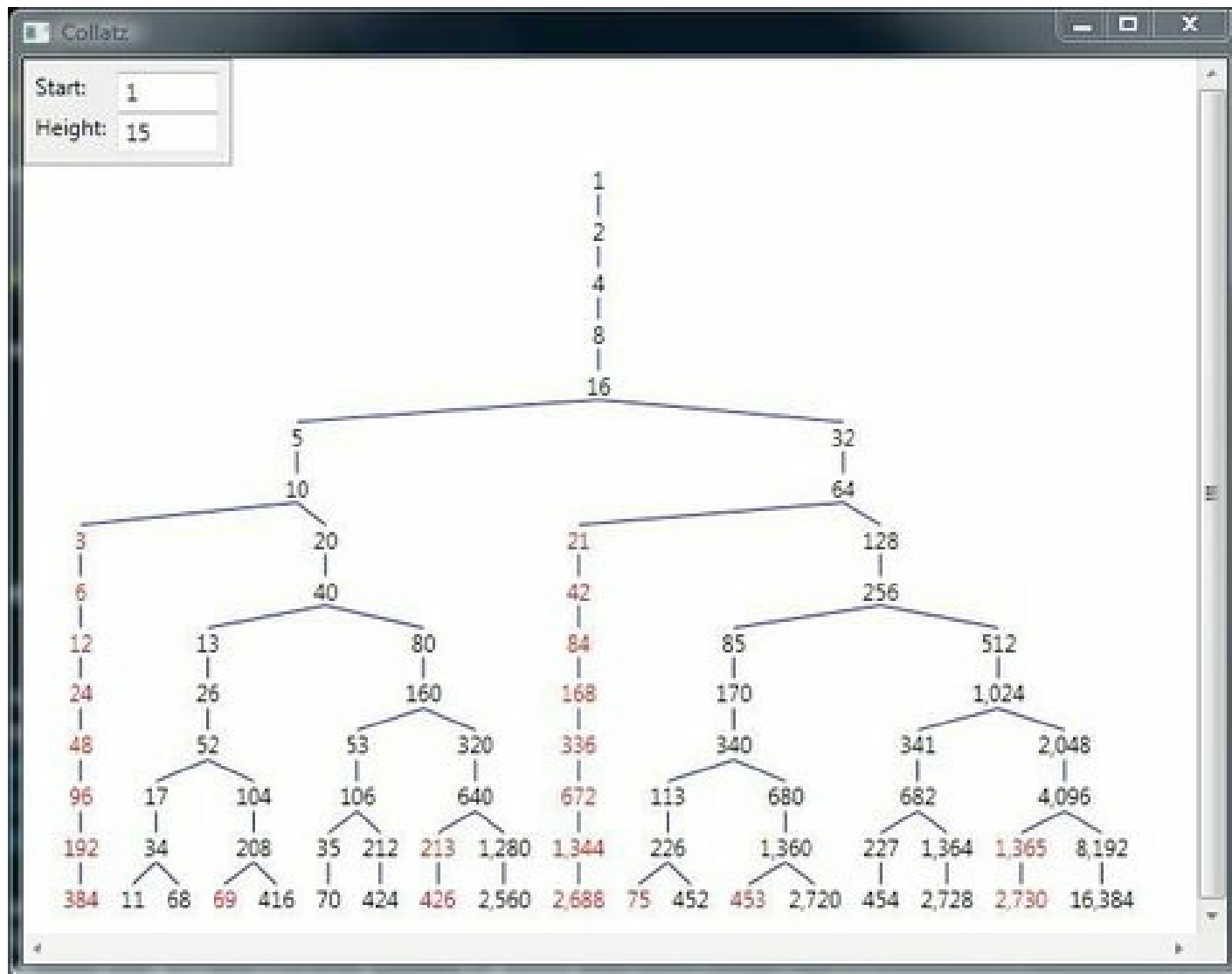
## A Few Mathematical Answers

Every starting point greater than  $2^K$  must take at least  $K$  steps to reach 1.

No number in the sequence can appear more than once (or else there's a loop).

The paths of the Collatz sequences form a tree that connects every number  $N$  to the root value 1.

# A bit of the Collatz Tree



# Exercise

What happens if we replace " $3n+1$ " by " $3n-1$ " and look at that sequence? (Start at 5)

What happens if we replace " $3n+1$ " by " $5n-1$ " and look at that sequence? (Start at 7)

Loops and infinite sequences NEVER (seem to) happen with the Collatz  $3n+1$  rule!



# The Value of Computation

Paul Erdos (every mathematician should learn about him!) said "Mathematics may not be ready for such problems," but he offered \$500 for a solution.

Computational methods let us see this complicated structure, appreciate its patterns, and make conjectures (mathematical guesses) about what is going on.

Perhaps the riddles of the Collatz sequence will be solved in the mathematical future.

## Leading up to Fibonacci

Before we examine our second sequence for which stepping stones are the right description, we have to learn how to handle slightly more complicated stepping stone rules. Once we see how to deal with a "two step" rule, we can proceed to:

### **Case 2: The Fibonacci Sequence**

# Example: The Triangular Sequence

We could write a *MATLAB* program that only stores one number, the current value of "a(i)".

$a(0): a \leftarrow 0$	$\leftarrow$ Initial condition
$a(1): a \leftarrow a + 1 = 0 + 1 = 1$	$\leftarrow$ Step $i$ adds $i$ to $a$
$a(2): a \leftarrow a + 2 = 1 + 2 = 3$	
$a(3): a \leftarrow a + 3 = 3 + 3 = 6$	
$a(4): a \leftarrow a + 4 = 6 + 4 = 10$	
$a(5): a \leftarrow a + 5 = 10 + 5 = 15$	
$a(6): a \leftarrow a + 6 = 15 + 6 = 21$	

Maybe temporarily keep 1 previous value

Sometimes it's convenient to keep old and new values, "aold" and "a":

$i=0: \quad a = 1$

$i=1: \text{ aold} = a, a = \text{aold} + i$

$i=2: \text{ aold} = a, a = \text{aold} + i$

$i=3: \text{ aold} = a, a = \text{aold} + i$

Useful to detect convergence, for instance.

# triangular\_keep1.m

```
n = input ( 'Enter maximum index of element: ' )
```

```
a = 0;
```

```
for i = 0 : n
```

```
    if ( i == 0 )
```

```
        a = 0;
```

```
    else
```

```
        aold = a;
```

```
        a = aold + i;
```

```
    end
```

```
    fprintf ( ' a(%d) = %d\n', n, a );
```

```
end
```

# The Derangement Sequence

A stepping stone rule that needs TWO old values:

$$d(0) = 1;$$

$$d(1) = 0;$$

$$d(i) = (i-1) * ( d(i-1) + d(i-2) )$$

$D(N)$  counts the ways to arrange numbers 1 through  $N$  so that no number is in the "right" position.

$$D(0) = 1 \text{ (mathematical convention)}$$

$$D(1) = 0$$

$$D(2) = 1 \text{ ( "21" )}$$

$$D(3) = 2 \text{ ( "231", "312" )}$$

$$D(4) = 9 \text{ ( "2143", "2341", "2413", ... )}$$

# Program a double stepping stone?

Assuming we start at index 0, we have to remember that there are now **two** special initializing steps,  $i=0$  and  $i=1$ ;

We need to store two previous values, maybe **dold** and **doldest**.

Sequence values  $d(0)$  through  $d(13)$  are:

1, 0, 1, 2, 9, 44, 265, 1854, 14833, 133496,  
1334961, 14684570, 176214841,  
2290792932

# derangement\_keep2.m

```
n = input ( 'Enter the element index: ' );

for i = 0 : n

    if ( i == 0 )      ← set D
        d = 1;
    elseif ( i == 1 ) ← remember DOLD, and set D
        dold = d;
        d = 0;
    else              ← remember DOLDEST, DOLD, and set D
        doldest = dold;
        dold = d;
        d = ( i - 1 ) * ( dold + doldest );
    end

    fprintf ( 'D(%d) = %d\n', i, d );

end
```



# The habit of rabbits

A puzzle from the middle ages:

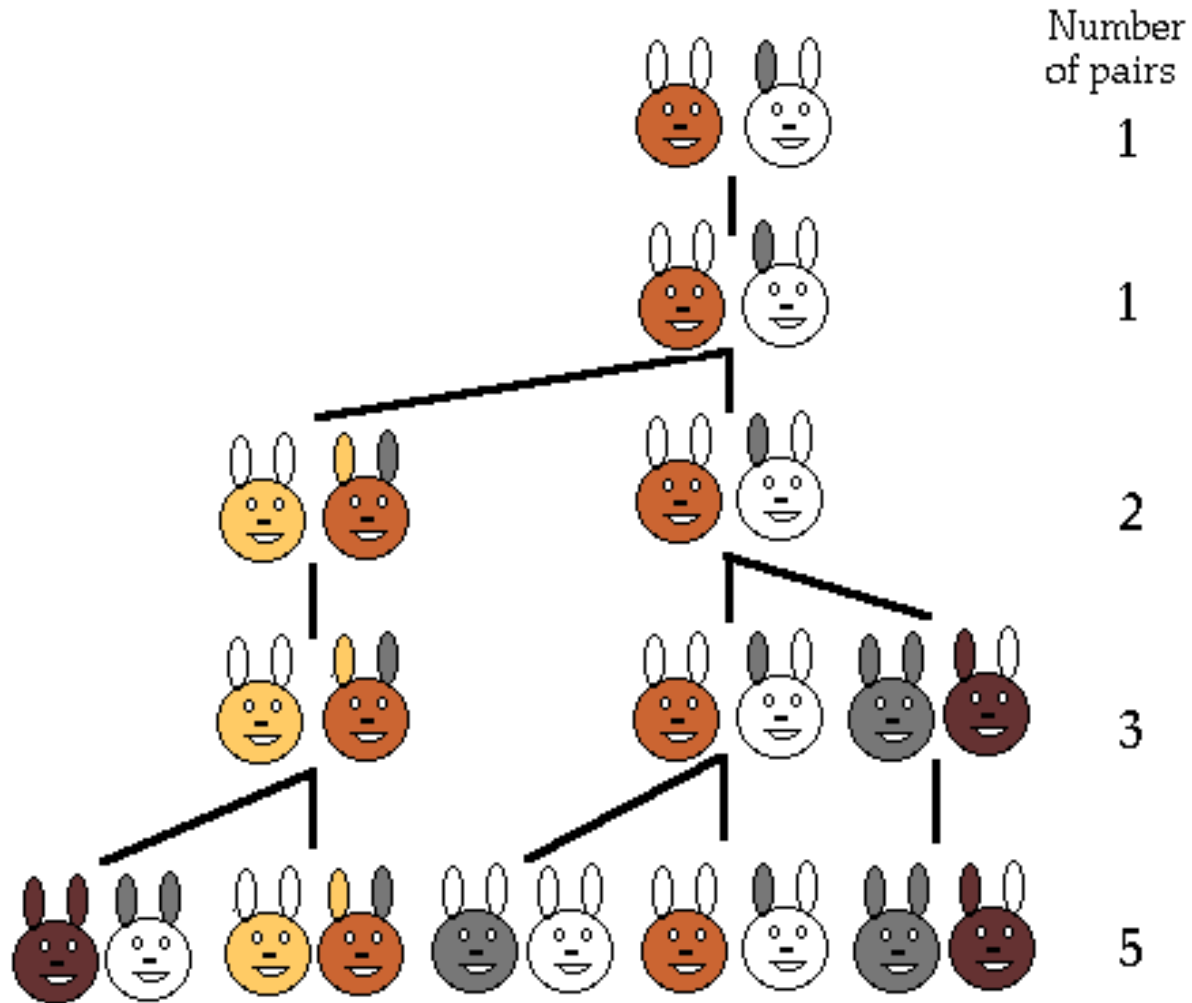
A child is given a pair of newborn rabbits.

These rabbits are "mathematical"

because, one month after they are born, the rabbits can mate, and one month after that, another breeding pair is born.

And so on, and so on. Which means what, exactly?

# First Five Months



# Formula vs Stepping Stone

The sequence counting the pairs of rabbits begins 1,1,2,3,5,8,13,21,34,55,89,144...

It's not easy to see a formula for these numbers (it turns out there is one), but if you just look at the numbers, you can see a stepping stone rule:

to get  $a(i)$ , just add  $a(i-1)$  and  $a(i-2)$ .

# fibonacci\_keep2.m

```
n = input ( 'Enter the element index: ' );

for i = 0 : n

    if ( i == 0 )      ← set F
        f = 1;
    elseif ( i == 1 ) ← remember FOLD, and set F
        fold = f;
        f = 1;
    else              ← remember FOLDEST, FOLD, and set F
        foldest = fold;
        fold = f;
        f = fold + foldest;
    end

    fprintf ( 'F(%d) = %d\n', i, f );

end
```

# A Fibonacci Formula

Although there's no formula for entries in the Collatz sequence, there is one for the Fibonacci sequence...but it's surprisingly complicated.

You could find out where this formula comes from if you take a class in difference equations (NOT differential equations!).

# The Fibonacci Formula

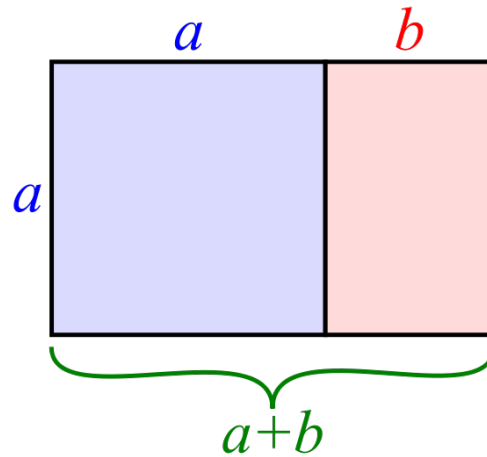
$$F(n) = \frac{\frac{1+\sqrt{5}}{2}^n - \frac{1-\sqrt{5}}{2}^n}{\sqrt{5}}$$

# fibonacci\_formula.m

```
% fibonacci_formula.m
% Exact formula for the N-th Fibonacci number.
%
n = input ( 'Enter the index N: ' );

f = ( ...
    ( ( 1 + sqrt ( 5 ) ) / 2 ) ^ n ...
    - ( ( 1 - sqrt ( 5 ) ) / 2 ) ^ n ...
    ) ...
    / sqrt ( 5 );
%
% MATLAB regards F as a real number, so we have to ROUND() it
% so that the %d format will print it nicely.
%
fprintf ( ' F(%d) = %d\n', n, round ( f ) );
```

# Golden Rectangle, Ratio, Proportion



A "golden" rectangle has the property that adding a square to the long side makes a larger rectangle, but in the same "golden ratio" of about 1.6 to 1.



# Fibonacci Ratio → Golden Ratio

The ratios of Fibonacci numbers converge to the golden ratio, whose value is exactly  $(1+\sqrt{5})/2$  or 1.61803...

$$1/1 = 1$$

$$2/1 = 2$$

$$3/2 = 1.5$$

$$5/3 = 1.66\dots$$

$$8/5 = 1.6$$

$$13/8 = 1.6250$$

# fibonacci\_ratio

```
n = input ( 'Enter the maximum element index: ' );
```

```
for i = 0 : n
```

```
    if ( i == 0 )
```

```
        f = 1;
```

```
    elseif ( i == 1 )
```

```
        fold = f;
```

```
        f = 1;
```

```
    else
```

```
        foldest = fold;
```

```
        fold = f;
```

```
        f = fold + foldest;
```

```
    end
```

```
    if ( 0 < i )
```

```
        fprintf ( 'R(%d) = %20.16g\n', i, f / fold );
```

```
    end
```

```
end
```

Insight Through

## Two References:

Brian Hayes, "On the ups and downs of the hailstone numbers",  
[hayes\_hailstone.pdf]

Martin Gardner, "The multiple fascinations of the Fibonacci sequence",  
[gardner\_1969\_03.pdf]

# In Class Exercise

For the Collatz sequence, we replace each odd number  $n$  by  $3n+1$ . As far as we know, this sequence has no loops (unless you count  $4 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ), and every starting point eventually decreases to 1.

Consider the rule where we replace odd  $n$  by  $3n-1$ . Then for the starting point  $n=5$  we have  $5 \rightarrow 14 \rightarrow 7 \rightarrow 20 \rightarrow 10 \rightarrow 5$ , which means we have entered an infinite loop.

If we replace odd  $n$  by  $5n+1$ , then for the starting point 7 we seem to get an sequence that blows up.

Collatz was very smart or very lucky to study the  $3n+1$  version of this process!

# Exam on Thursday

There is a practice exam available on Canvas, in the "files" directory for "09\_28".

The actual exam will be very similar.

There will NOT be questions on plotting, or on `fprintf`, `rand` or `randi` or MATLAB lists.

You should review `mod()`, `gcd()`, `&&`, `||`, `for`, `if/elseif/if`, `while(true)`, `while(condition)`.

You should take your time on the questions, and think step by step.