Lecture 9: Recurrence Relations

Matthew Fricke

July 15, 2013

This Lecture

1. Definition
2. Examples
3. Guess and Check
4. Binary Search
5. Characteristic Equation Method
6. The Fibonacci Sequence
7. Golden Ratio
8. Gambler's Ruin

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.
- Recurrence relations are closely tied to differential equations because they are both self referential.

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.
- Recurrence relations are closely tied to differential equations because they are both self referential.
- A differential equation relates a function to its own derivative.

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.
- Recurrence relations are closely tied to differential equations because they are both self referential.
- A differential equation relates a function to its own derivative.
- The discrete version of a differential equation is a difference equation.

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.
- Recurrence relations are closely tied to differential equations because they are both self referential.
- A differential equation relates a function to its own derivative.
- The discrete version of a differential equation is a difference equation.
- Sometimes people call recurrence relations difference equations but really difference equations are just a type of recurrence relation.

What are Recurrence Relations?

- A recurrence relation is an equation that defines a value in a sequence using previous values in the sequence.
- Recurrence relations are closely tied to differential equations because they are both self referential.
- A differential equation relates a function to its own derivative.
- The discrete version of a differential equation is a difference equation.
- Sometimes people call recurrence relations difference equations but really difference equations are just a type of recurrence relation.
- Some of the techniques for solving recurrence relations are almost the same as those used to solve differential equations.

Examples of Recursion Relations

Towers of Hanoi

$$T_0 = 0$$
$$T_n = 2T_{n-1} + 1$$

Fibonacci Sequence

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}$$

Compound Interest

Stirling Numbers

Towers of Hanoi

- Recall that the recurrence T_n represents the amount of work needed to solve the problem.

Towers of Hanoi

- Recall that the recurrence T_n represents the amount of work needed to solve the problem.
- We solved the problem by figuring out a general statement characterizing the amount of work needed to move *one* disk from one peg to another: $2(n-1)+1$ where n is the number of disks.

Towers of Hanoi

- Recall that the recurrence T_n represents the amount of work needed to solve the problem.
- We solved the problem by figuring out a general statement characterizing the amount of work needed to move *one* disk from one peg to another: $2(n-1)+1$ where n is the number of disks.
- Then we wrote out the sequence values for $T_0, T_1, T_2, T_3, T_4 \ldots = 0, 1, 3, 7, 15 \ldots$

Towers of Hanoi

- Recall that the recurrence T_n represents the amount of work needed to solve the problem.
- We solved the problem by figuring out a general statement characterizing the amount of work needed to move *one* disk from one peg to another: $2(n-1)+1$ where n is the number of disks.
- Then we wrote out the sequence values for $T_0, T_1, T_2, T_3, T_4 \ldots = 0, 1, 3, 7, 15 \ldots$
- Guessed that the formula was $2^n - 1$

Towers of Hanoi

- Recall that the recurrence T_n represents the amount of work needed to solve the problem.
- We solved the problem by figuring out a general statement characterizing the amount of work needed to move *one* disk from one peg to another: $2(n-1)+1$ where n is the number of disks.
- Then we wrote out the sequence values for $T_0, T_1, T_2, T_3, T_4 \ldots = 0, 1, 3, 7, 15 \ldots$
- Guessed that the formula was $2^n - 1$
- ... and proved it with mathematical induction.

Binary Search Example

- We are going to come up with an algorithm to solve a search problem.

Binary Search Example

- We are going to come up with an algorithm to solve a search problem.
- then analyse the amount of work needed to search using recurrence relations.

Binary Search Example

- We are going to come up with an algorithm to solve a search problem.
- then analyse the amount of work needed to search using recurrence relations.
- Given a *sorted* array of integers how might we determine if a particular integer is in the array?

Binary Search Example

```
function BinSearch(key, array, start, end)
    if end ≤ start then
        return FALSE
    else
        mid ← (end+start)/2
        if key = array[mid] then
            return TRUE
        else if key ≤ array[mid] then
            return BinSearch(key, array, start, mid)
        else if key > array[mid] then
            return BinSearch(key, array, mid, end)
```

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.
- We are thinking about the work done in the worst case (we might find the key on the first try).

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.
- We are thinking about the work done in the worst case (we might find the key on the first try).
- Typically "work" means operations or memory accesses.

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.
- We are thinking about the work done in the worst case (we might find the key on the first try).
- Typically "work" means operations or memory accesses.
- Notice that due to Line 5 in the algorithm each recursive call only needs to work on half the array that the previous call worked on.

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.
- We are thinking about the work done in the worst case (we might find the key on the first try).
- Typically "work" means operations or memory accesses.
- Notice that due to Line 5 in the algorithm each recursive call only needs to work on half the array that the previous call worked on.
- So we guess that $T(n) = T(\frac{n}{2}) + 1, n =$ the size of the array.

Binary Search Example

- Let's define a recurrence relation $T(n)$ (T for time) that describes the amount of work to be done by BinarySearch.
- We are thinking about the work done in the worst case (we might find the key on the first try).
- Typically "work" means operations or memory accesses.
- Notice that due to Line 5 in the algorithm each recursive call only needs to work on half the array that the previous call worked on.
- So we guess that $T(n) = T(\frac{n}{2}) + 1, n =$ the size of the array.
- Base case: $T(2) = 1$

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.
- We recognize this as 2^n which is an exponential function.

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.
- We recognize this as 2^n which is an exponential function.
- If we halve the values at each step $\ldots 16, 8, 4, 2, 1$ we are doing the inverse.

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.
- We recognize this as 2^n which is an exponential function.
- If we halve the values at each step $\ldots 16, 8, 4, 2, 1$ we are doing the inverse.
- The inverse of an exponential function is the logarithmic function.

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.
- We recognize this as 2^n which is an exponential function.
- If we halve the values at each step $\ldots 16, 8, 4, 2, 1$ we are doing the inverse.
- The inverse of an exponential function is the logarithmic function.
- Therefore we guess that that $T(n) = \log_2(n)$

Binary Search Example

- We would like to put the recurrence in closed form. That is we would like to solve the recurrence.
- Intuitively if we have a sequence that doubles at each step we get: $1, 2, 4, 8, 16 \ldots$.
- We recognize this as 2^n which is an exponential function.
- If we halve the values at each step $\ldots 16, 8, 4, 2, 1$ we are doing the inverse.
- The inverse of an exponential function is the logarithmic function.
- Therefore we guess that that $T(n) = \log_2(n)$
- Base case: $T(2) = 1$

Binary Search Example

- The next step is to try and prove our guess was right with mathematical induction.

Binary Search Example

- The next step is to try and prove our guess was right with mathematical induction.
- Proof by strong induction:

Binary Search Example

- The next step is to try and prove our guess was right with mathematical induction.
- Proof by strong induction:
- Base case $n = 2$: $\log_2(2 \cdot 2) = 1$, QED for Base Case

Binary Search Example

- The next step is to try and prove our guess was right with mathematical induction.
- Proof by strong induction:
- Base case $n = 2$: $\log_2(2 \cdot 2) = 1$, QED for Base Case
- Inductive Step: $\forall j < k$, $T(j) = T(\frac{j}{2}) + 1 = \log_2(j) \implies T(k) = T(\frac{k}{2}) + 1 = \log_2(k)$

Binary Search Example

- Proof of the inductive step:

Binary Search Example

- Proof of the inductive step:
- $T(k) = T(\frac{k}{2}) + 1$ this is a premise.

Binary Search Example

- Proof of the inductive step:
- $T(k) = T(\frac{k}{2}) + 1$ this is a premise.
- $\frac{k}{2} < k \implies T(\frac{k}{2}) = \log_2(\frac{k}{2})$ induction hypothesis.

Binary Search Example

- Proof of the inductive step:
- $T(k) = T(\frac{k}{2}) + 1$ this is a premise.
- $\frac{k}{2} < k \implies T(\frac{k}{2}) = \log_2(\frac{k}{2})$ induction hypothesis.
- $T(\frac{k}{2}) = \log_2 \frac{k}{2} = \log_2 k - \log_2 2 = \log_2 k - 1$ by log rules.

Binary Search Example

- Proof of the inductive step:
- $T(k) = T(\frac{k}{2}) + 1$ this is a premise.
- $\frac{k}{2} < k \implies T(\frac{k}{2}) = \log_2(\frac{k}{2})$ induction hypothesis.
- $T(\frac{k}{2}) = \log_2 \frac{k}{2} = \log_2 k - \log_2 2 = \log_2 k - 1$ by log rules.
- $T(k) = T(\frac{k}{2}) + 1 = \{\log_2(k) - 1\} + 1$ by substitution.

Binary Search Example

- Proof of the inductive step:
- $T(k) = T(\frac{k}{2}) + 1$ this is a premise.
- $\frac{k}{2} < k \implies T(\frac{k}{2}) = \log_2(\frac{k}{2})$ induction hypothesis.
- $T(\frac{k}{2}) = \log_2 \frac{k}{2} = \log_2 k - \log_2 2 = \log_2 k - 1$ by log rules.
- $T(k) = T(\frac{k}{2}) + 1 = \{\log_2(k) - 1\} + 1$ by substitution.
- $T(k) = \log_2(k)$ by simplification of $-1 + 1$. QED

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:
- Assume what we want is true. Then $T(k) = \log_2(k)$

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:
- Assume what we want is true. Then $T(k) = \log_2(k)$
- We want to make $T(k) = T(k/2) + 1 = \log_2(x)$

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:
- Assume what we want is true. Then $T(k) = \log_2(k)$
- We want to make $T(k) = T(k/2) + 1 = \log_2(x)$
- We have that it is true for everything less than k by strong induction.

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:
- Assume what we want is true. Then $T(k) = \log_2(k)$
- We want to make $T(k) = T(k/2) + 1 = \log_2(x)$
- We have that it is true for everything less than k by strong induction.
- So solve for $x : \log_2(x) = \log_2(k) - 1 \implies \log_2(x) - \log_2(2) \implies \log_2(x) = \log_2(k/2)$

Binary Search Example

- Proofs are often presented in the opposite order to which they were developed.
- The progression of thought for this proof is:
- Assume what we want is true. Then $T(k) = \log_2(k)$
- We want to make $T(k) = T(k/2) + 1 = \log_2(x)$
- We have that it is true for everything less than k by strong induction.
- So solve for $x : \log_2(x) = \log_2(k) - 1 \implies \log_2(x) - \log_2(2) \implies \log_2(x) = \log_2(k/2)$
- $\therefore x = k/2$ Since $k/2$ is less than k we can use proof by strong induction.

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.
- More analytical methods include The Master Method, Recursion Trees, and Annihilators.

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.
- More analytical methods include The Master Method, Recursion Trees, and Annihilators.
- We will learn a method using the *Characteristic Equation* of a recurrence.

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.
- More analytical methods include The Master Method, Recursion Trees, and Annihilators.
- We will learn a method using the *Characteristic Equation* of a recurrence.
- (The term Characteristic Equation comes from Linear Algebra)

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.
- More analytical methods include The Master Method, Recursion Trees, and Annihilators.
- We will learn a method using the *Characteristic Equation* of a recurrence.
- (The term Characteristic Equation comes from Linear Algebra)
- The method works on second-order linear recurrence relations with constant coefficients. Annihilators generalize this method to any order.

Characteristic Equation Method

- Guess-and-Check is a very common approach (variations are method of iteration and substitution). Most useful when you already have enough experience and intuition to be able to look at a recurrence and know the answer from comparison with other similar problems.
- More analytical methods include The Master Method, Recursion Trees, and Annihilators.
- We will learn a method using the *Characteristic Equation* of a recurrence.
- (The term Characteristic Equation comes from Linear Algebra)
- The method works on second-order linear recurrence relations with constant coefficients. Annihilators generalize this method to any order.
- Second order recurrence: refers to two previous values of the recurrence, e.g. $T_n = T_{n-1} + T_{n-2}$

Determining the Characteristic Equation

Definition: A second-order linear homogeneous recurrence relation with constant coefficients is a recurrence relation of the form:
$a_k = A \cdot a_{k-1} + B \cdot a_{k-2}, \exists m \in \mathbb{Z} \ni \forall k \in \mathbb{Z}, k \geq m$, where A and B are fixed real numbers with $B \neq 0$.

(The part with m and k just allows for there to be some base cases.)

Determining the Characteristic Equation

- A second-order linear homogeneous recurrence relation is satisfied by the sequence $1, t, t^2, t^3, t^4 \ldots, t^n$

Determining the Characteristic Equation

- A second-order linear homogeneous recurrence relation is satisfied by the sequence $1, t, t^2, t^3, t^4 \ldots, t^n$
- i.e. $t^k = A \cdot t^{k-1} + B \cdot t^{t-2}$ because each term is equal to A times the previous term plus B times the term before that.

Determining the Characteristic Equation

- A second-order linear homogeneous recurrence relation is satisfied by the sequence $1, t, t^2, t^3, t^4 \ldots, t^n$
- i.e. $t^k = A \cdot t^{k-1} + B \cdot t^{t-2}$ because each term is equal to A times the previous term plus B times the term before that.
- Dividing by t^{k-2} gives $t^2 - At - B = 0$.

Determining the Characteristic Equation

- A second-order linear homogeneous recurrence relation is satisfied by the sequence $1, t, t^2, t^3, t^4 \ldots, t^n$
- i.e. $t^k = A \cdot t^{k-1} + B \cdot t^{t-2}$ because each term is equal to A times the previous term plus B times the term before that.
- Dividing by t^{k-2} gives $t^2 - At - B = 0$.
- This is true in general. A sequence of the form t^k only satisfies the 2nd order homogeneous recurrence iff it satisfies $t^2 - At - B = 0$.

Determining the Characteristic Equation

- A second-order linear homogeneous recurrence relation is satisfied by the sequence $1, t, t^2, t^3, t^4 \ldots, t^n$
- i.e. $t^k = A \cdot t^{k-1} + B \cdot t^{t-2}$ because each term is equal to A times the previous term plus B times the term before that.
- Dividing by t^{k-2} gives $t^2 - At - B = 0$.
- This is true in general. A sequence of the form t^k only satisfies the 2nd order homogeneous recurrence iff it satisfies $t^2 - At - B = 0$.
- We call this equation the Characteristic Equation.

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- This is of the form $a_k = 1 \cdot a_{k-1} + B \cdot a_{k-2}$ where $A = 1$, and $B = 2$

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- This is of the form $a_k = 1 \cdot a_{k-1} + B \cdot a_{k-2}$ where $A = 1$, and $B = 2$
- So the characteristic equation $t^2 - At - B = 0$ is $t^2 - t - 2 = 0$.

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- This is of the form $a_k = 1 \cdot a_{k-1} + B \cdot a_{k-2}$ where $A = 1$, and $B = 2$
- So the characteristic equation $t^2 - At - B = 0$ is $t^2 - t - 2 = 0$.
- Now we need to find t so that the characteristic equation is satisfied, i.e. equal to zero.

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- This is of the form $a_k = 1 \cdot a_{k-1} + B \cdot a_{k-2}$ where $A = 1$, and $B = 2$
- So the characteristic equation $t^2 - At - B = 0$ is $t^2 - t - 2 = 0$.
- Now we need to find t so that the characteristic equation is satisfied, i.e. equal to zero.
- Recall all the fun you had finding roots of quadratic formulas in Algebra I!

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- $t^2 - t - 2 = (t-2)(t+1)$

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- $t^2 - t - 2 = (t-2)(t+1)$
- The roots are 2 and -1.

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- $t^2 - t - 2 = (t-2)(t+1)$
- The roots are 2 and -1.
- So the only sequences of the form t^n that satisfy the recurrence are:
  $r_n = 2^0, 2^1, 2^2, 2^3, 2^4 \ldots$ and
  $s_n = (-1)^0, (-1)^1, (-1)^2, (-1)^3, (-1)^4 \ldots$

Example

- Find a sequence that satisfies $a_k = a_{k-1} + 2a_{k-2}$
- $t^2 - t - 2 = (t-2)(t+1)$
- The roots are 2 and -1.
- So the only sequences of the form t^n that satisfy the recurrence are:
  $r_n = 2^0, 2^1, 2^2, 2^3, 2^4 \ldots$ and
  $s_n = (-1)^0, (-1)^1, (-1)^2, (-1)^3, (-1)^4 \ldots$
- AND any linear combination of these sequences satisfies the recurrence: $a_n = C \cdot r_n + D \cdot s_n$, C and D can be any numbers.

Distinct Roots Theorem

All that leads us to the Distinct Roots Theorem. Suppose a sequence $a_0, a_1, a_2, \ldots$ satisfies a recurrence relation $a_k = A \cdot a_{k-1} + B \cdot a_{k-1}$ for some real numbers A and B and k > 2.
Then $a_0, a_1, a_2, \ldots$ satisfies the closed form $a_n = C \cdot r^n + D \cdot s^n$ if the characteristic equation $t^2 - At - B = 0$ has two distinct roots r and s. Where C and D are solutions to the system of equations $a_0 = C \cdot r^0 + D \cdot s^0$ and $a_1 = C \cdot r^1 + D \cdot s^1$.
Equivalently $a_0 = C + D$ and $a_1 = C \cdot r + D \cdot s$

Single Root Theorem

and the Single Root Theorem. Suppose a sequence $a_0, a_1, a_2, \ldots$ satisfies a recurrence relation $a_k = A \cdot a_{k-1} + B \cdot a_{k-1}$ for some real numbers A and B and k > 2.

Then $a_0, a_1, a_2, \ldots$ satisfies the closed form $a_n = C \cdot r^n + nD \cdot r^n$ if the characteristic equation $t^2 - At - B = 0$ has a single (perhaps repeated) root r. Where C and D are solutions to the system of equations $a_0 = C \cdot r^0 + nD \cdot r^0$ and $a_1 = C \cdot r^1 + nD \cdot r^1$. Equivalently $a_0 = C + D$ and $a_1 = C \cdot r + nD \cdot r$.

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.
- Recall the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, with base cases $F_0 = 0, F_1 = 1$.

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.
- Recall the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, with base cases $F_0 = 0, F_1 = 1$.
- The Fibonacci sequence is a second-order and homogeneous with constant coefficients: A=1, B=1

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.
- Recall the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, with base cases $F_0 = 0, F_1 = 1$.
- The Fibonacci sequence is a second-order and homogeneous with constant coefficients: A=1, B=1
- The characteristic equation is $t^2 - t - 1 = 0$

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.
- Recall the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, with base cases $F_0 = 0, F_1 = 1$.
- The Fibonacci sequence is a second-order and homogeneous with constant coefficients: A=1, B=1
- The characteristic equation is $t^2 - t - 1 = 0$
- Solving this equation for t:
  $t = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$ for any quadratic polynomial:
  $at^2 + b^t + c = 0$ (Quadratic Formula).

Fibonacci Sequence

- Now we are equipped to solve recurrences like the Fibonacci sequence.
- Recall the Fibonacci sequence is $F_n = F_{n-1} + F_{n-2}$, with base cases $F_0 = 0, F_1 = 1$.
- The Fibonacci sequence is a second-order and homogeneous with constant coefficients: A=1, B=1
- The characteristic equation is $t^2 - t - 1 = 0$
- Solving this equation for t:
  $t = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}$ for any quadratic polynomial:
  $at^2 + b^t + c = 0$ (Quadratic Formula).
- $t = \frac{1 \pm \sqrt{1 - 4(-1)}}{2} = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots.

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.
- so by the distinct root theorem:
  $F_n = C\left(\frac{1+\sqrt{5}}{2}\right)^n + D\left(\frac{1-\sqrt{5}}{2}\right)^n$

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.
- so by the distinct root theorem:
  $F_n = C\left(\frac{1+\sqrt{5}}{2}\right)^n + D\left(\frac{1-\sqrt{5}}{2}\right)^n$
- Now we need to find the values of C and D using the initial conditions (differential equations), base cases (recurrences).

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.
- so by the distinct root theorem:
  $F_n = C\left(\frac{1+\sqrt{5}}{2}\right)^n + D\left(\frac{1-\sqrt{5}}{2}\right)^n$
- Now we need to find the values of C and D using the initial conditions (differential equations), base cases (recurrences).
- The base cases are $F_0 = 0$ and $F_1 = 1$.

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.
- so by the distinct root theorem:
  $F_n = C\left(\frac{1+\sqrt{5}}{2}\right)^n + D\left(\frac{1-\sqrt{5}}{2}\right)^n$
- Now we need to find the values of C and D using the initial conditions (differential equations), base cases (recurrences).
- The base cases are $F_0 = 0$ and $F_1 = 1$.
- So we need to solve the system of equations:
  $F_1 = 1 = C + D$
  $F_2 = 1 = C\rho_1 + D\rho_2$

Fibonacci Sequence

- $t = \frac{1 \pm \sqrt{5}}{2}$. Two distinct roots, ρ_1 and ρ_2.
- so by the distinct root theorem:
  $F_n = C\left(\frac{1+\sqrt{5}}{2}\right)^n + D\left(\frac{1-\sqrt{5}}{2}\right)^n$
- Now we need to find the values of C and D using the initial conditions (differential equations), base cases (recurrences).
- The base cases are $F_0 = 0$ and $F_1 = 1$.
- So we need to solve the system of equations:
  $F_1 = 1 = C + D$
  $F_2 = 1 = C\rho_1 + D\rho_2$
- Solving this system gives $C = \frac{1+\sqrt{5}}{2\sqrt{5}}, D = \frac{-1+\sqrt{5}}{2\sqrt{5}}$ One way to solve this system using the method of partial fractions. If you know this method the setup is:
  $F(z) = \frac{z}{1-z-z^2} = \frac{1}{\sqrt{5}}\left(\frac{1}{1-\rho_1 z} - \frac{1}{1-\rho_2 z}\right)$

Fibonacci Sequence

- So we have solved the Fibonacci recurrence:

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

Fibonacci Sequence

- So we have solved the Fibonacci recurrence:

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

- Seems strange that this sequence is made up of integers but we have $\sqrt{5}$ throughout.

Fibonacci Sequence

- So we have solved the Fibonacci recurrence:

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

- Seems strange that this sequence is made up of integers but we have $\sqrt{5}$ throughout.
- It turns out that this value $\frac{1+\sqrt{5}}{2}$ is very special. It is called the Golden Ratio or Golden Mean and has the symbol Φ

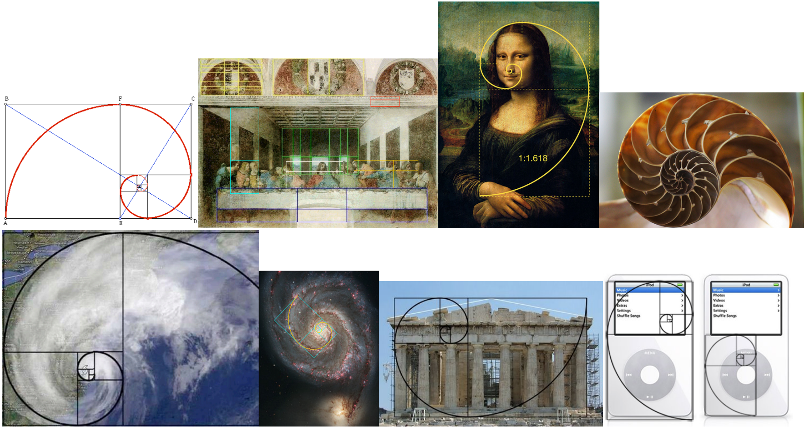
Fibonacci Sequence

- So we have solved the Fibonacci recurrence:

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

- Seems strange that this sequence is made up of integers but we have $\sqrt{5}$ throughout.
- It turns out that this value $\frac{1+\sqrt{5}}{2}$ is very special. It is called the Golden Ratio or Golden Mean and has the symbol Φ
- During the Renaissance Φ was known as the Divine Proportion

The Golden Ratio



Gambler's Ruin

Consider a game of chance. You (the player) will win \$1 or lose \$1 depending on the outcome of a coin toss. If the coin comes up heads you win if it comes up tails you lose.

You decide to play until one of two conditions are met:
1) You run out of money.
2) or you have won a target amount of money, M.

The question we would like to answer is the probability of you going bust given a starting amount of money and the target value, M.

Gambler's Ruin

- Notice that the amount of money you have in the first round is equal to your starting amount.
- Otherwise the amount of money you have depends on the amount of money you had previous round coupled with the outcome of the previous coin toss.
- So we have a sequence of values for the probability of going bust P_n given \$$n$ that depends on previous values of P_k.
- In other words we have a recurrence that we can try to define and solve.

Gambler's Ruin

- The probability of a player having \$$k$, i.e. P_k transitioning into P_{k-1} (losing a dollar) is $\frac{1}{2}$.
- The probability of transitioning to state P_{k+1} (winning a dollar) is also $\frac{1}{2}$.
- These outcomes are related by xor so we can use the addition rule of discrete probability.
- Therefore $P_k = \frac{1}{2}P_{k-1} + \frac{1}{2}P_{k+1}$.

Gambler's Ruin

- Therefore $P_k = \frac{1}{2}P_{k-1} + \frac{1}{2}P_{k+1}$.
- Rewriting in a form we are more used to: $0 = -2P_k + P_{k-1} + P_{k+1}$.
- Having the coefficient 2 on P_k is awkward but we can shift the sequence index by -1.
- $0 = -2P_{k-1} + P_{k-2} + P_k$.
- Rewriting: $P_k = 2P_{k-1} - P_{k-2}$.

Gambler's Ruin

- Now we have a second order homogeneous recurrence with constant coefficients.
- We know how to solve those using the Characteristic Equation method.
- What about the base cases though. Here the sequence ends under two circumstances:
- The player wins a total of $\$M$ or the player loses all their money.
- So the base cases (more like boundary conditions in this case) are $P_M = 0$ and $P_0 = 1$.
- Since the Probability of going bust when you have \$0 is 1. The Probability of going bust when you have $\$M$ is 0.

Gambler's Ruin

- Now we have everything we need to solve the recurrence.
- Characteristic Equation: $t^2 - 2t + 1 = 0$ since $A = 2$ and $B = -1$ in $P_k = AP_{k-1} + BP_{k-2}$ and the characteristic equation is $t^2 - At + B = 0$.
- Now we find the roots of the characteristic polynomial.
- Factoring: $(t-1)(t-1)$ The repeated root is $\rho = 1$.
- Using the single root theorem: $P_n = C(1)^n + nD(1)^n$.
- Solving for C and D:
  $P_0 = 1 = C + (0)D$
  $P_M = 0 = C + (M)D$
  $\therefore C = 1$
  $\therefore D = -\frac{1}{M}$
  $\therefore P_n = 1 - n\frac{1}{M}$
  $\therefore P_n = \frac{M}{M} - n\frac{1}{M} = \frac{M-n}{M}$

Gambler's Ruin

- Now we have a closed form for the recurrence and can answer the question for any target amount and starting amount of money.
- Example: What is the probability of going bust if you start with \$30 and your goal is \$120.
- Answer: $M = 120, n = 30$. $P_30 = \frac{120-30}{120} = 75\%$.
- Another Examples: $M = 500, n = 50$. $P_50 = \frac{500-50}{500} = 90\%$ chance of going bust before winning \$500.