Recursive Definitions

And Applications to Counting

C(n,k)

- $C(n,k) = C(n-1,k-1) + C(n-1,k)$ (where $n,k \geq 1$)
  - Easy derivation: Let $|S|=n$ and $a \in S$.

    $C(n,k) =$ # k-sized subsets of S containing a

    \+ # k-sized subsets of S not containing a

- In fact, gives a recursive definition of C(n,k)
  - Base case (to define for $k \leq n$): $C(n,0) = C(n,n) = 1$ for all $n \in \mathbb{N}$
  - Or, to define it for all $(n,k) \in \mathbb{N} \times \mathbb{N}$

    Base case: $C(n,0)=1$, for all $n \in \mathbb{N}$, and $C(0,k)=0$ for all $k \in \mathbb{Z}^+$

n \ k	0	1	2	3	4	5	6
0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	2	1	0	0	0	0
3	1	3	3	1	0	0	0
4	1	4	6	4	1	0	0
5	1	5	10	10	5	1	0
6	1	6	15	20	15	6	1

Tower of Hanoi

- Move entire stack of disks to another peg
  - Move one from the top of one stack to the top of another
  - A disk cannot be placed on top of a smaller disk
- How many moves needed?
- Optimal number not known when 4 pegs and over ≈30 disks!
- Optimal solution known for 3 pegs (and any number of disks)

Tower of Hanoi

- Recursive algorithm (optimal for 3 pegs)
  - Transfer(n,A,C):

    If n=1, move the single disk from peg A to peg C

    Else

        Transfer(n-1,A,B) (leaving the largest disk out of play)

        Move largest disk to peg C

        Transfer(n-1,B,C) (leaving the largest disk out of play)

Tower of Hanoi

- Recursive algorithm (optimal for 3 pegs)
  - Transfer(n,A,C):

    If n=1, move the single disk from peg A to peg C

    Else

    Transfer(n-1,A,B) (leaving the largest disk out of play)

    Move largest disk to peg C

    Transfer(n-1,B,C) (leaving the largest disk out of play)
- How many moves are made by this algorithm?
- $M(n)$ be the number of moves made by the above algorithm
- $M(n) = 2M(n-1) + 1$ with $M(1) = 1$
- 1, 3, 7, 15, 31, ...

Recursive Definitions

- E.g., $f(0) = 1$ (Initial Condition)
  $f(n) = n \cdot f(n-1) \quad \forall n \in \mathbb{Z}$ s.t. $n>0$ (Recurrence relation)

- $f(n) = n \cdot (n-1) \cdot \ldots \cdot 1 \cdot 1 = n!$

- This is the formal definition of n!

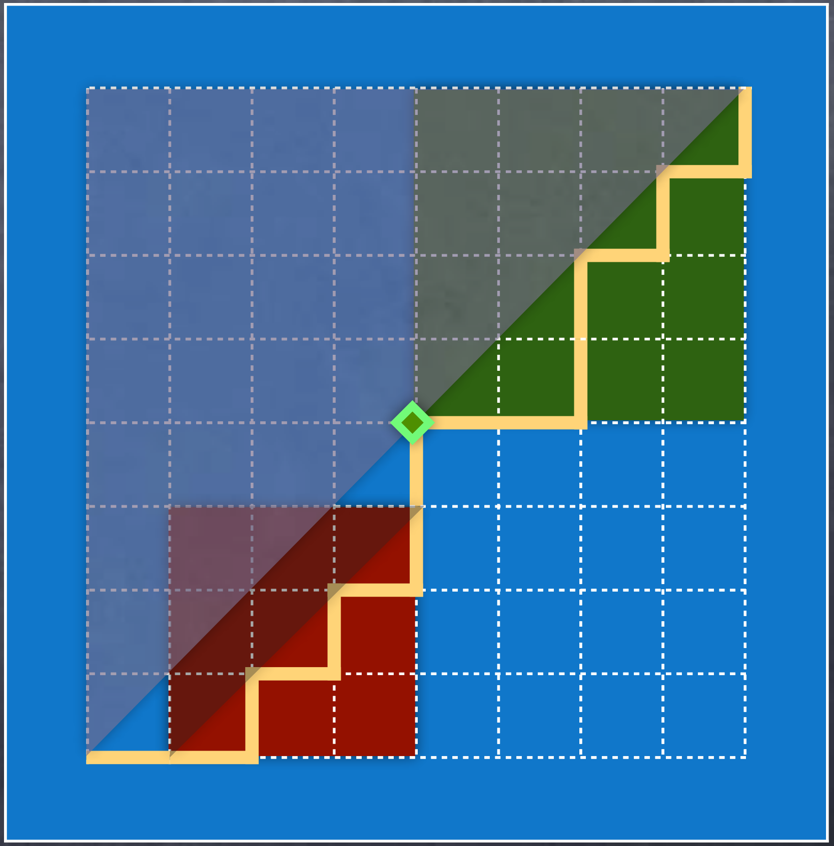
- Translates to a program to compute factorial:

```
factorial(n ∈ ℕ)  {
   if (n==0) return 1;
   else return n*factorial(n-1);
}
```

```
factorial(n ∈ ℕ)  {
   F[0] = 1;
   for i in 1..n
      F[i] = i*F[i-1];
   return F[n];
}
```

Catalan Numbers

- How many paths are there in the grid from (0,0) to (n,n) without ever crossing over to the y>x region?
- Any path can be constructed as follows
  - Pick minimum k>0 s.t. (k,k) reached
  - $(0,0) \rightarrow (1,0) \Rrightarrow (k,k-1) \rightarrow (k,k) \Rrightarrow (n,n)$ where $\Rrightarrow$ denotes a Catalan path
- $\mathrm{Cat}(n) = \Sigma_{k=1 \text{ to } n} \mathrm{Cat}(k-1) \cdot \mathrm{Cat}(n-k)$
- $\mathrm{Cat}(0) = 1$
- 1, 1, 2, 5, 14, 42, 132, ...



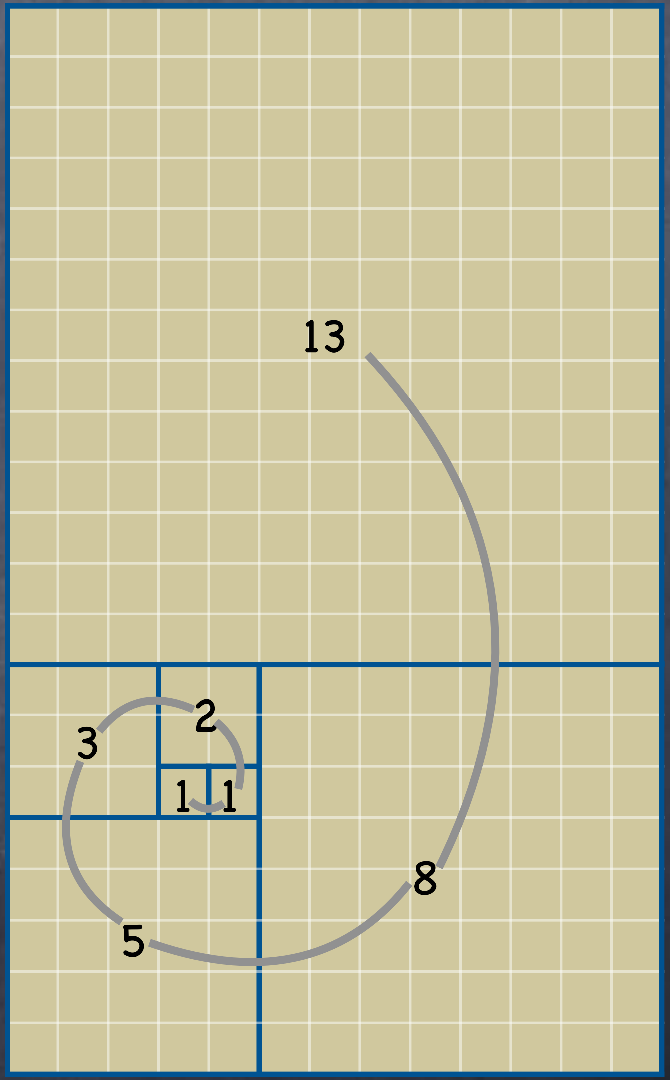
e.g., $42 = 1 \cdot 14 + 1 \cdot 5 + 2 \cdot 2 + 5 \cdot 1 + 14 \cdot 1$

Closed form expression? Later

Fibonacci Sequence

- $F(0) = 0$
  $F(1) = 1$
  $F(n) = F(n-1) + F(n-2) \quad \forall n \geq 2$
- $F(n)$ is the n^{th} Fibonacci number (starting with 0^{th})

Closed form expression? Coming up



Counting Strings

- How many ternary strings of length n which don't have "00" as a substring?
- Set up a recurrence
  - $A(n)$ = # such strings starting with 0
  - $B(n)$ = # such strings not starting with 0
  - $A(n) = B(n-1)$. $B(n) = 2(A(n-1) + B(n-1))$. [Why?]
- Initial condition: $A(0) = 0$; $B(0) = 1$ (empty string)
- Required count: $A(n) + B(n)$
- Can rewrite in terms of just B
  - $B(0) = 1$. $B(1) = 2$. $B(n) = 2B(n-1) + 2B(n-2) \quad \forall n \geq 2$
  - Required count: $B(n-1) + B(n)$.

Recursion & Induction

- Claim: $F(3n)$ is even, where $F(n)$ is the n^{th} Fibonacci number, $\forall n \geq 0$

0 1 1 **2** 3 5 **8** 13 21 **34**...
Stronger claim (but easier to prove by induction): $F(n)$ is even iff n is a multiple of 3

- Proof by induction:
- Base case:
  $n=0$: $F(3n) = F(0) = 0$ ✔ $n=1$: $F(3n) = F(3) = 2$ ✔
- Induction step: for all $k \geq 2$
  Induction hypothesis: suppose for $0 \leq n \leq k-1$, $F(3n)$ is even
  To prove: $F(3k)$ is even
  - $F(3k) = F(3k-1) + F(3k-2) = ?$
    - Unroll further: $F(3k-1) = F(3k-2) + F(3k-3)$
      $F(3k) = 2 \cdot F(3k-2) + F(3(k-1))$ = even, by induction hypothesis

Closed Form

- Sometimes possible to get a "closed form" expression for a quantity defined recursively (in terms of simpler operations)
  - e.g., $f(0)=0$ & $f(n) = f(n-1) + n, \forall n>0$
    - $f(n) = n(n+1)/2$
- Sometimes, we just give it a name
  - e.g., $n!$, Fibonacci(n), Cat(n)
  - In fact, formal definitions of integers, addition, multiplication etc. are recursive
    - e.g., $0 \cdot a = 0$ & $n \cdot a = (n-1) \cdot a + a, \forall n>0$
    - e.g., $2^0 = 1$ & $2^n = 2 \cdot 2^{n-1}$
- Sometimes both
  - e.g., Fibonacci(n), Cat(n) have closed forms

Closed Form via Induction

- $f(0) = c.\quad f(1) = d.\quad f(n) = a\cdot f(n-1) + b\cdot f(n-2) \quad \forall n \geq 2.$

Exercise: Fibonacci numbers

- Suppose $X^2 - aX - b = 0$ has two distinct (possibly complex) solutions, x and y

Characteristic equation: replace f(n) by X^n in the recurrence

- Claim: $\exists p,q\ \forall n\ f(n) = p\cdot x^n + q\cdot y^n$
- Let $p=(d-cy)/(x-y)$, $q=(d-cx)/(y-x)$ so that base cases n=0,1 work
- Inductive step: for all $k \geq 2$
  Induction hypothesis: $\forall n$ s.t. $1 \leq n \leq k-1$, $f(n) = px^n + qy^n$
  To prove: $f(k) = px^k + qy^k$
  - $$\begin{aligned} f(k) &= a\cdot f(k-1) + b\cdot f(k-2) \\ &= a\cdot(px^{k-1}+qy^{k-1}) + b\cdot(px^{k-2}+qy^{k-2}) - px^k - qy^k + px^k + qy^k \\ &= -px^{k-2}(x^2-ax-b) - qy^{k-2}(y^2-ay-b) + px^k + qy^k = px^k + qy^k \ \checkmark \end{aligned}$$

Closed Form via Induction

- $f(0) = c.$ $f(1) = d.$ $f(n) = a \cdot f(n-1) + b \cdot f(n-2)$ $\forall n \geq 2.$
- Suppose $X^2 - aX - b = 0$ has only one solution $x \neq 0$
  i.e., $X^2 - aX - b = (X-x)^2$, or equivalently, $a=2x$, $b=-x^2$
- Claim: $\exists p,q\ \forall n\ f(n) = (p + q \cdot n)x^n$
- Let $p = c$, $q = d/x-c$ so that base cases $n=0,1$ work
- Inductive step: for all $k \geq 2$
  Induction hypothesis: $\forall n$ s.t. $1 \leq n \leq k-1$, $f(n) = (p + qn)y^n$
  To prove: $f(k) = (p+qk)x^k$
  - $f(k) = a \cdot f(k-1) + b \cdot f(k-2)$
    $= a\ (p+qk-q)x^{k-1} + b \cdot (p+qk-2q)x^{k-2} - (p+qk)x^k + (p+qk)x^k$
    $= -(p+qk)x^{k-2}(x^2-ax-b) - qx^{k-2}(ax+2b) + (p+qk)x^k = (p+qk)x^k$ ✓

Solving a Recurrence

- Often, once a correct guess is made, easy to prove by induction
- How does one guess?
- Will see a couple of approaches
  - By unrolling the recurrence into a chain or a "rooted tree"
  - Using the "method of generating functions"