CS 105: Department Introductory Course on Discrete Structures

Instructor : S. Akshay

Sep 26, 2025

Lecture 14 – Counting and Combinatorics

Logistics

Midsem corrections over now: Check your moodle.

- Extra help sessions on BASIC topics this saturday.
- Please follow piazza and ask, if necessary.
- Students with $< 18/50$ are strongly encouraged to attend.

Counting and Combinatorics

Topics to be covered

- Basics of counting
  - Product principle
  - Sum principle
  - Bijection principle
  - Double counting
- Subsets, partitions, Permutations and combinations
  1. Binomial coefficients and Binomial theorem
  2. Pascal's triangle
  3. Permutations and combinations with repetitions

Counting and Combinatorics

Topics to be covered

- Basics of counting
  - Product principle
  - Sum principle
  - Bijection principle
  - Double counting
- Subsets, partitions, Permutations and combinations
  1. Binomial coefficients and Binomial theorem
  2. Pascal's triangle
  3. Permutations and combinations with repetitions
  4. Estimating $n!$

Counting and Combinatorics

Topics to be covered

- Basics of counting
  - Product principle
  - Sum principle
  - Bijection principle
  - Double counting
- Subsets, partitions, Permutations and combinations
  1. Binomial coefficients and Binomial theorem
  2. Pascal's triangle
  3. Permutations and combinations with repetitions
  4. Estimating $n!$
- Today: Recurrence relations and generating functions

Estimating $n!$

How big is $n!$?

- It is clearly bigger than n and n^2.

Estimating $n!$

How big is $n!$?

- It is clearly bigger than n and n^2.
- Is it bigger than 2^n, n^n?

Estimating $n!$

How big is $n!$?

- It is clearly bigger than n and n^2.
- Is it bigger than 2^n, n^n?
- Easy to see: for all $n \geq 4$,

$$2^n \leq n! \leq n^n$$

Estimating $n!$

How big is $n!$?

- It is clearly bigger than n and n^2.
- Is it bigger than 2^n, n^n?
- Easy to see: for all $n \geq 4$,

$$2^n \leq n! \leq n^n$$

- Can we do better?
  Can we quantify how much more n^n is compared to $n!$?

Estimating $n!$

How big is $n!$?

- It is clearly bigger than n and n^2.
- Is it bigger than 2^n, n^n?
- Easy to see: for all $n \geq 4$,

$$2^n \leq n! \leq n^n$$

- Can we do better?
  Can we quantify how much more n^n is compared to $n!$?

Theorem (Stirling's Approximation)

$$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$$

where e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$.

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

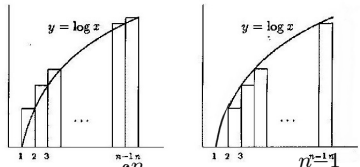
Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

Now, we relate it to natural log function as shown in the figure.



$$\sum_{i=1}^{n} \log(i) \geq \int_{1}^{n} \log(x)dx \geq \sum_{i=1}^{n} \log(i)$$

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

$$\sum_{i=1}^{n} \log(i) \geq \int_1^n \log(x)dx \geq \sum_{i=1}^{n-1} \log(i)$$

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

$$\sum_{i=1}^{n} \log(i) \geq \int_1^n \log(x)dx \geq \sum_{i=1}^{n-1} \log(i)$$

$$S \geq x\log(x) - x|_1^n \geq S - \log(n)$$

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

$$\sum_{i=1}^{n} \log(i) \geq \int_1^n \log(x)dx \geq \sum_{i=1}^{n-1} \log(i)$$

$$S \geq x\log(x) - x|_1^n \geq S - \log(n)$$

$$S \geq n\log(n) - n + 1 \geq S - \log(n).$$

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

$$\sum_{i=1}^{n} \log(i) \geq \int_1^n \log(x)dx \geq \sum_{i=1}^{n-1} \log(i)$$

$$S \geq x\log(x) - x|_1^n \geq S - \log(n)$$

$$S \geq n\log(n) - n + 1 \geq S - \log(n).$$

Raising both sides to power of e we get

- l.h.s. $n! \geq e^{n\log(n)-n+1} = (n/e)^n e$ and

Approximating the factorial

Theorem (Stirling's Approximation)

$e\left(\frac{n}{e}\right)^n \leq n! \leq ne\left(\frac{n}{e}\right)^n$ where, e is the base of natural logarithms, $\log(e) = e^{\log(e)} = 1$

Proof: Let $S = \log(n!) = \sum_{i=1}^{n} \log(i)$. Thus, $e^S = n!$

$$\sum_{i=1}^{n} \log(i) \geq \int_1^n \log(x)dx \geq \sum_{i=1}^{n-1} \log(i)$$

$$S \geq x\log(x) - x|_1^n \geq S - \log(n)$$

$$S \geq n\log(n) - n + 1 \geq S - \log(n).$$

Raising both sides to power of e we get

- l.h.s. $n! \geq e^{n\log(n)-n+1} = (n/e)^n e$ and
- r.h.s. $n! \leq e^{(n+1)\log(n)-n+1} = n^{n+1}/e^{n-1} = ne(n/e)^n$.

□

Next: Recurrence relations and generating functions

Recall: No. of subsets of a set of n elements

How many subsets does a set A of n elements have?

- Induction
- Product principle: two choices for each element, hence $2 \cdot 2 \cdots 2 \cdot 2$ (n-times).
- Bijection: between $\mathcal{P}(X)$ and n-length $\{0,1\}$-sequences.
- Sum principle: Subsets of size 0 + subsets of size 1 + ... + subsets of size n = Total number of subsets.

Next: Recurrence relations and generating functions

Recall: No. of subsets of a set of n elements

How many subsets does a set A of n elements have?

- Induction
- Product principle: two choices for each element, hence $2 \cdot 2 \cdots 2 \cdot 2$ (n-times).
- Bijection: between $\mathcal{P}(X)$ and n-length $\{0,1\}$-sequences.
- Sum principle: Subsets of size 0 + subsets of size 1 + ... + subsets of size n = Total number of subsets.
- Recurrence: $F(n) = 2 \cdot F(n-1), F(0) = 1$.

Next: Recurrence relations and generating functions

Recall: No. of subsets of a set of n elements

How many subsets does a set A of n elements have?

- Induction
- Product principle: two choices for each element, hence $2 \cdot 2 \cdots 2 \cdot 2$ (n-times).
- Bijection: between $\mathcal{P}(X)$ and n-length $\{0,1\}$-sequences.
- Sum principle: Subsets of size 0 + subsets of size 1 + ... + subsets of size n = Total number of subsets.
- Recurrence: $F(n) = 2 \cdot F(n-1), F(0) = 1$.

Next: Recurrence relations and generating functions

Recall: No. of subsets of a set of n elements

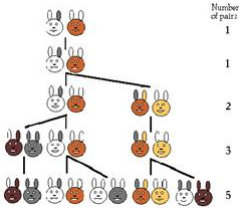
How many subsets does a set A of n elements have?

- Induction
- Product principle: two choices for each element, hence $2 \cdot 2 \cdots 2 \cdot 2$ (n-times).
- Bijection: between $\mathcal{P}(X)$ and n-length $\{0,1\}$-sequences.
- Sum principle: Subsets of size 0 + subsets of size 1 + ... + subsets of size n = Total number of subsets.
- Recurrence: $F(n) = 2 \cdot F(n-1), F(0) = 1$.

How do we form recurrences and how do we solve them?

Another example of recurrence: The Fibonacci Sequence

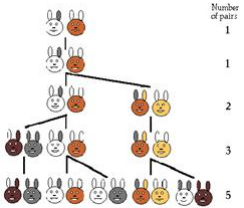




- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$

Another example of recurrence: The Fibonacci Sequence

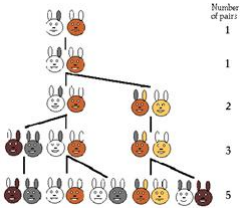




- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$
- Recurrence relation: $u_n = u_{n-1} + u_{n-2}$ where $u_1 = u_0 = 1$

Another example of recurrence: The Fibonacci Sequence

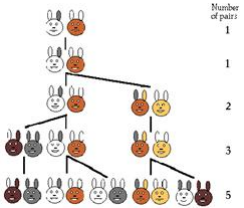




- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$
- Recurrence relation: $u_n = u_{n-1} + u_{n-2}$ where $u_1 = u_0 = 1$
- But rabbits die!

Another example of recurrence: The Fibonacci Sequence





- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$
- Recurrence relation: $u_n = u_{n-1} + u_{n-2}$ where $u_1 = u_0 = 1$
- But rabbits die!
- Consider $u_n = u_{n-1} + u_{n-2} - u_{n-3}$ where $u_2 = 2, u_1 = u_0 = 1$

Recurrence and linear recurrence relations

Definition

- A recurrence relation for a sequence is an equation that expresses its n^{th} term using one or more of the previous terms of the sequence.

Recurrence and linear recurrence relations

Definition

- A recurrence relation for a sequence is an equation that expresses its n^{th} term using one or more of the previous terms of the sequence.
- A linear recurrence relation is of the form

$$u_n = a_{k-1}u_{n-1} + \ldots + a_1 u_{n-k+1} + a_0 u_{n-k}$$

where $a_0, \ldots, a_{k-1} \in \mathbb{R}, k \in \mathbb{N}$ are constants.

Recurrence and linear recurrence relations

Definition

- A recurrence relation for a sequence is an equation that expresses its n^{th} term using one or more of the previous terms of the sequence.
- A linear recurrence relation is of the form

$$u_n = a_{k-1}u_{n-1} + \ldots + a_1 u_{n-k+1} + a_0 u_{n-k}$$

  where $a_0, \ldots, a_{k-1} \in \mathbb{R}, k \in \mathbb{N}$ are constants.

- k is called the degree/depth/order of the sequence.

Recurrence and linear recurrence relations

Definition

- A recurrence relation for a sequence is an equation that expresses its n^{th} term using one or more of the previous terms of the sequence.
- A linear recurrence relation is of the form

$$u_n = a_{k-1}u_{n-1} + \ldots + a_1 u_{n-k+1} + a_0 u_{n-k}$$

where $a_0, \ldots, a_{k-1} \in \mathbb{R}, k \in \mathbb{N}$ are constants.

- k is called the degree/depth/order of the sequence.
- The first few (e.g., k elements $u_0, \ldots, u_{k-1}$) are called initial conditions

Recurrence and linear recurrence relations

Definition

- A recurrence relation for a sequence is an equation that expresses its n^{th} term using one or more of the previous terms of the sequence.
- A linear recurrence relation is of the form

$$u_n = a_{k-1}u_{n-1} + \ldots + a_1 u_{n-k+1} + a_0 u_{n-k}$$

  where $a_0, \ldots, a_{k-1} \in \mathbb{R}, k \in \mathbb{N}$ are constants.

- k is called the degree/depth/order of the sequence.
- The first few (e.g., k elements $u_0, \ldots, u_{k-1}$) are called initial conditions
- Once the initial conditions and recurrence relation are given, the entire sequence is determined!

Applications of recurrences

Analysis of running time of algorithms

- How many comparisons is done by a binary search algorithm on a sorted list?

Applications of recurrences

Analysis of running time of algorithms

- How many comparisons is done by a binary search algorithm on a sorted list?

Mathematical Biology

- Population dynamics – rabbits, but also cell population!
- Viral spread.

Applications of recurrences

Analysis of running time of algorithms

- How many comparisons is done by a binary search algorithm on a sorted list?

Mathematical Biology

- Population dynamics – rabbits, but also cell population!
- Viral spread.

Digital Signal Processing

- Modeling feedback in a system; outputs at time i become inputs at time $i + 1$.

Applications of recurrences

Analysis of running time of algorithms

- How many comparisons is done by a binary search algorithm on a sorted list?

Mathematical Biology

- Population dynamics – rabbits, but also cell population!
- Viral spread.

Digital Signal Processing

- Modeling feedback in a system; outputs at time i become inputs at time $i + 1$.

Macroeconomics

- Time series data
- Mathematical models of economic systems

Exercise!

How many bit strings of length n are there having no two consecutive 0's?

1. How many such bit strings are there of length 2, 3, 4?
2. Find a recurrence relation for this.
3. Give the initial conditions
4. How many such bit strings are there of length 8?

Some more examples of recurrences

How many bit strings of length n are there having no two consecutive 0's?

1. How many such bit strings are there of length 2, 3, 4?
2. Find a recurrence relation for this.
3. Give the initial conditions
4. How many such bit strings are there of length 8?

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

Some more examples of recurrences

How many bit strings of length n are there having no two consecutive 0's?

1. How many such bit strings are there of length 2, 3, 4?
2. Find a recurrence relation for this.
3. Give the initial conditions
4. How many such bit strings are there of length 8?

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Example: $n = 3 : ((a + b) + c), (a + (b + c))$

Some more examples of recurrences

How many bit strings of length n are there having no two consecutive 0's?

1. How many such bit strings are there of length 2, 3, 4?
2. Find a recurrence relation for this.
3. Give the initial conditions
4. How many such bit strings are there of length 8?

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Example: $n = 3 : ((a + b) + c), (a + (b + c))$
- $n = 4 : (((a + b) + c) + d), ((a + b) + (c + d)), ((a + (b + c)) + d), ...$

Some more examples of recurrences

How many bit strings of length n are there having no two consecutive 0's?

1. How many such bit strings are there of length 2, 3, 4?
2. Find a recurrence relation for this.
3. Give the initial conditions
4. How many such bit strings are there of length 8?

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Example: $n = 3 : ((a + b) + c), (a + (b + c))$
- $n = 4 : (((a + b) + c) + d), ((a + b) + (c + d)), ((a + (b + c)) + d), ...$ In general, let $C(n)$ be the number of ways of doing this.

Examples of recurrences (Contd.)

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Let $C(n)$ be the number of ways of doing this.

Examples of recurrences (Contd.)

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Let $C(n)$ be the number of ways of doing this.
- If outermost bracketing $(A + B)$ appears between x_k and x_{k+1}, then there are $C(k) \cdot C(n - k)$ ways of bracketing it.

Examples of recurrences (Contd.)

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Let $C(n)$ be the number of ways of doing this.
- If outermost bracketing $(A + B)$ appears between x_k and x_{k+1}, then there are $C(k) \cdot C(n - k)$ ways of bracketing it.
- k can be anything from 1 till $n - 1$

Examples of recurrences (Contd.)

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Let $C(n)$ be the number of ways of doing this.
- If outermost bracketing $(A + B)$ appears between x_k and x_{k+1}, then there are $C(k) \cdot C(n-k)$ ways of bracketing it.
- k can be anything from 1 till $n-1$
- Thus, $C(n) = \sum_{i=1}^{n-1} C(i)C(n-i)$ for $n > 1$

Examples of recurrences (Contd.)

How many ways are there to bracket a sum of n terms so that it can be computed by adding two numbers at a time?

- Let $C(n)$ be the number of ways of doing this.
- If outermost bracketing $(A + B)$ appears between x_k and x_{k+1}, then there are $C(k) \cdot C(n-k)$ ways of bracketing it.
- k can be anything from 1 till $n-1$
- Thus, $C(n) = \sum_{i=1}^{n-1} C(i)C(n-i)$ for $n > 1$
- Initial conditions are $C(0) = 0$, $C(1) = 1$ (by convention).
- Note that $C(2) = 1, C(3) = 2, C(4) = 5$.
- This sequence are called Catalan numbers...

How do we solve such recurrences? We start with the Fibonacci sequence.

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

1. This is a linear recurrence relation...

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

1. This is a linear recurrence relation... $\therefore$ if we find any solutions, we can take linear combinations.

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

1. This is a linear recurrence relation... $\therefore$ if we find any solutions, we can take linear combinations.
2. i.e., if $F_n = F_{n-1} + F_{n-2}$, $G_n = G_{n-1} + G_{n-2}$ and $H_n = aF_n + bG_n$, then what is a recurrence for H_n?

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

1. This is a linear recurrence relation... $\therefore$ if we find any solutions, we can take linear combinations.
2. i.e., if $F_n = F_{n-1} + F_{n-2}$, $G_n = G_{n-1} + G_{n-2}$ and $H_n = aF_n + bG_n$, then $H_n = H_{n-1} + H_{n-2}$

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

1. This is a linear recurrence relation... $\therefore$ if we find any solutions, we can take linear combinations.
2. i.e., if $F_n = F_{n-1} + F_{n-2}$, $G_n = G_{n-1} + G_{n-2}$ and $H_n = aF_n + bG_n$, then $H_n = H_{n-1} + H_{n-2}$

Proof method 1 (for linear recurrences: try α^n!)

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it!

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it! $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it! $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$
5. Thus, general solution is $F_n = a(\frac{1+\sqrt{5}}{2})^n + b(\frac{1-\sqrt{5}}{2})^n$.

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it! $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$
5. Thus, general solution is $F_n = a(\frac{1+\sqrt{5}}{2})^n + b(\frac{1-\sqrt{5}}{2})^n$.
6. How do we get a and b?

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it! $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$
5. Thus, general solution is $F_n = a(\frac{1+\sqrt{5}}{2})^n + b(\frac{1-\sqrt{5}}{2})^n$.
6. Use F_0 and F_1 – initial conditions: $a = \frac{\sqrt{5}+1}{2\sqrt{5}}$, $b = \frac{\sqrt{5}-1}{2\sqrt{5}}$

How to "solve" the Fibonacci recurrence?

By solving, we mean give a closed-form expression for n^{th} term.

Fibonacci recurrence relation

For $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$, $F_0 = F_1 = 1$.

Proof method 1 (for linear recurrences: try α^n!)

1. try $F_n = \alpha^n$...
2. $\alpha^n = \alpha^{n-1} + \alpha^{n-2}$ implies $\alpha^{n-2}(\alpha^2 - \alpha - 1) = 0$.
3. So if $\alpha^2 - \alpha - 1 = 0$, the recurrence holds for all n.
4. Solve it! $\alpha = \frac{1+\sqrt{5}}{2}, \beta = \frac{1-\sqrt{5}}{2}$
5. Thus, general solution is $F_n = a(\frac{1+\sqrt{5}}{2})^n + b(\frac{1-\sqrt{5}}{2})^n$.
6. Use F_0 and F_1 – initial conditions: $a = \frac{\sqrt{5}+1}{2\sqrt{5}}$, $b = \frac{\sqrt{5}-1}{2\sqrt{5}}$

Thus, $F_n = \frac{\sqrt{5}+1}{2\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n + \frac{\sqrt{5}-1}{2\sqrt{5}}(\frac{1-\sqrt{5}}{2})^n$.

Solving general linear recurrence sequences

- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$?

Solving general linear recurrence sequences

- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$? Is $a_{25} > 2^{25}$?

Solving general linear recurrence sequences

- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$? Is $a_{25} > 2^{25}$?
- Find a solution for $a_n = 6a_{n-1} - 9a_{n-2}$, with $a_0 = 1, a_1 = 6$? Can you apply the same method for this?

Solving general linear recurrence sequences

- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$? Is $a_{25} > 2^{25}$?
- Find a solution for $a_n = 6a_{n-1} - 9a_{n-2}$, with $a_0 = 1, a_1 = 6$? Can you apply the same method for this? What went wrong?
- Recall the recurrence for Catalan Numbers: $C(n) = \sum_{i=1}^{n-1} C(i)C(n-i)$ for $n > 1$, $C(0) = 0$, $C(1) = 1$.
  No. of ways to bracket a sum of n terms s.t. it can be computed by adding two numbers at a time?

Solving general linear recurrence sequences

- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$? Is $a_{25} > 2^{25}$?
- Find a solution for $a_n = 6a_{n-1} - 9a_{n-2}$, with $a_0 = 1, a_1 = 6$? Can you apply the same method for this? What went wrong?
- Recall the recurrence for Catalan Numbers: $C(n) = \sum_{i=1}^{n-1} C(i)C(n-i)$ for $n > 1$, $C(0) = 0$, $C(1) = 1$.
  No. of ways to bracket a sum of n terms s.t. it can be computed by adding two numbers at a time?

This method does not work if we have repeated roots (this can be fixed!) and non-linear recurrences.

Solving general linear recurrence sequences

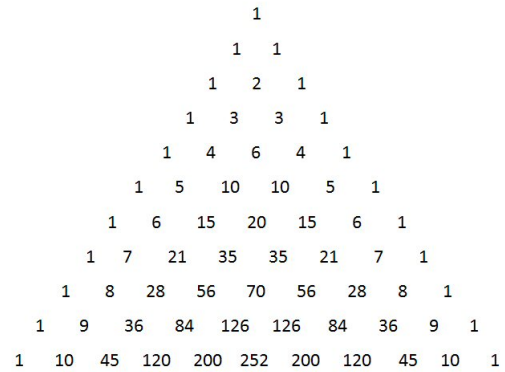
- (C.W) Find a solution for $a_n = a_{n-1} + 2a_{n-2}$ with $a_0 = 2$, $a_1 = 7$? Is $a_{25} > 2^{25}$?
- Find a solution for $a_n = 6a_{n-1} - 9a_{n-2}$, with $a_0 = 1, a_1 = 6$? Can you apply the same method for this? What went wrong?
- Recall the recurrence for Catalan Numbers: $C(n) = \sum_{i=1}^{n-1} C(i)C(n-i)$ for $n > 1$, $C(0) = 0$, $C(1) = 1$.
  No. of ways to bracket a sum of n terms s.t. it can be computed by adding two numbers at a time?

This method does not work if we have repeated roots (this can be fixed!) and non-linear recurrences.

Reading assignment

Read examples/generalizations from Sections 6.1 and 6.2 from Rosen's book (6th Edition).

An aside: find the Fibonacci sequence!



- $F(n) = F(n-1) + F(n-2)$.
- $1, 1, 2, 3, 5, 8, 13, ...$
- Can you observe the sum of which terms in the Pascal's triangle gives rise to the terms of the Fibonacci sequence?