CS 105: DIC on Discrete Structures

Instructor : S. Akshay

Sept 12, 2024

Lecture 20 – Counting and Combinatorics

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm
- Venue: LA 002, LH 102, LT 103, LT 105, LT 003

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm
- Venue: LA 002, LH 102, LT 103, LT 105, LT 003
- Duration: 2 hrs(+Compensatory Time for PwD students)

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm
- Venue: LA 002, LH 102, LT 103, LT 105, LT 003
- Duration: 2 hrs(+Compensatory Time for PwD students)

1. Be in the hall 15min *before* start of the exam.
2. Bring pen, id card, water bottle (and nothing else).

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm
- Venue: LA 002, LH 102, LT 103, LT 105, LT 003
- Duration: 2 hrs(+Compensatory Time for PwD students)

1. Be in the hall 15min *before* start of the exam.
2. Bring pen, id card, water bottle (and nothing else).

Syllabus

First 17 lectures of course, till & incl Thursday Sept 05.

Logistics: Midsem

CS105 Midsem Exam

- Date and Time: Sept 21st, Saturday, at 16:00 hrs, 4pm
- Venue: LA 002, LH 102, LT 103, LT 105, LT 003
- Duration: 2 hrs(+Compensatory Time for PwD students)

1. Be in the hall 15min *before* start of the exam.
2. Bring pen, id card, water bottle (and nothing else).

Syllabus

First 17 lectures of course, till & incl Thursday Sept 05.

- Basic reasoning: Propositions, proofs, induction,
- Basic structures: sets, functions, (un)countability, relations, posets, chains, anti-chains, lattices.
- Basic counting: counting principles, double counting, permutations & combinations.

Logistics

- Extra help sessions on different BASIC topics being conducted by TAs.
- Only Basic material, NOT advanced.
- Please follow piazza and ask, if necessary.

Logistics

- Extra help sessions on different BASIC topics being conducted by TAs.
- Only Basic material, NOT advanced.
- Please follow piazza and ask, if necessary.

Also,

- Pattern of exam similar to quiz, some easy/basic, some hard.

Logistics

- Extra help sessions on different BASIC topics being conducted by TAs.
- Only Basic material, NOT advanced.
- Please follow piazza and ask, if necessary.

Also,

- Pattern of exam similar to quiz, some easy/basic, some hard.
- Solve more questions from Kenneth Rosen etc.

Logistics

- Extra help sessions on different BASIC topics being conducted by TAs.
- Only Basic material, NOT advanced.
- Please follow piazza and ask, if necessary.

Also,

- Pattern of exam similar to quiz, some easy/basic, some hard.
- Solve more questions from Kenneth Rosen etc.
- Few more extra/advanced questions may be released (no solutions, but can discuss on piazza).

Counting and Combinatorics

Topics to be covered

- Basics of counting
  - Product principle
  - Sum principle
  - Bijection principle
  - Double counting

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 $n!$
- Recurrence relations and generating functions
- Pigeonhole Principle and its extensions

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

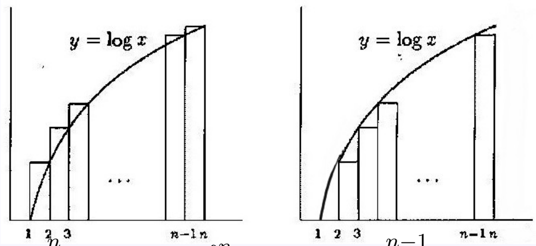
Theorem (Stirling's Approximation)

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

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.



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

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, $e^S = n!$

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

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)

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

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, $e^S = n!$

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

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

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

Raising both sides to power of e we get

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

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

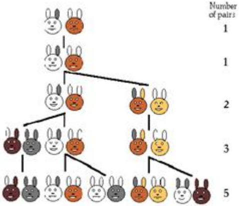
Recall: No. of subsets of a set of n elements

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: $F(n) = 2 \cdot F(n-1), F(0) = 1$.

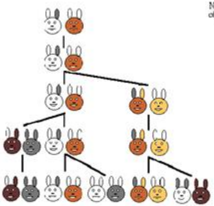
How do we form recurrences and how do we solve them?

Another example of recurrence: The Fibonacci Sequence



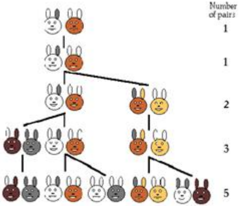
- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$

Another example of recurrence: The Fibonacci Sequence



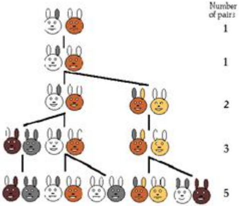
- Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots$
- Recurrence relation: $u_n = u_{n-1} + u_{n-2}$ where $u_1 = u_0 = 1$

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 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 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 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

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

Digital Signal Processing

- Modeling feedback in a syste; 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

Digital Signal Processing

- Modeling feedback in a syste; 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 that do not have two consecutive 0's?

- Find a recurrence relation for this
- Give the initial conditions
- How many such bit strings are there of length 7?

Some more examples of recurrences

How many bit strings of length n are there that do not have two consecutive 0's?

- Find a recurrence relation for this
- Give the initial conditions
- How many such bit strings are there of length 7?

An aside: find the Fibonacci sequence!

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

1 7 21 35 35 21 7 1

1 8 28 56 70 56 28 8 1

1 9 36 84 126 126 84 36 9 1

1 10 45 120 200 252 200 120 45 10 1

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