UCLA COURSE FIELD NOTES

MATH 61

Introduction to Discrete Structures

Logic, proof, sets, counting, relations, recurrences, graphs, and trees

How these notes are written. I wrote each chapter the way I wish the material had been explained the first time: the real idea, the point where I usually get stuck, the simplest mental model that still stays honest, and a worked decision instead of a polished answer appearing from nowhere. Every chapter closes with practice and a fully written solution. If a sentence sounds impressive but does not help me solve or explain something, it does not belong here.

Daniel Mastick

Curriculum map, working notes, practice, and solutions

Course map

Week	Core thread	What should be solid by the end
1	**Logic and quantifiers**	Propositions, implication, equivalence, predicates, and negation.
2	**Proof methods**	Direct, contrapositive, contradiction, cases, existence, and uniqueness.
3	**Sets and functions**	Set algebra, images, inverse images, injections, surjections, and cardinality.
4	**Induction**	Weak and strong induction; recursively defined objects and invariants.
5	**Counting**	Sum/product rules, permutations, combinations, and binomial coefficients.
6	**Advanced counting**	Pigeonhole principle, inclusion-exclusion, and combinatorial proofs.
7	**Relations**	Equivalence relations, partitions, partial orders, and Hasse diagrams.
8	**Recurrences**	Linear recurrences, characteristic roots, and recursive counting.
9	**Graphs**	Paths, cycles, degrees, connectivity, bipartite graphs, and Euler trails.
10	**Trees**	Equivalent definitions, spanning trees, traversal, and structural induction.

Scope anchor: https://catalog.registrar.ucla.edu/course/2025/MATH61. The sequence is aligned to the official catalog description and expanded into a practical ten-week study plan.

How I would use this packet

Treat each numbered section as one chapter. Read the formal explanation, pause at the student interpretation, and see whether the easy version still says the same thing. Rework the example without looking, then cover the solution in the chapter practice box. The cumulative set at the end is deliberately mixed: knowledge becomes durable when I have to choose the tool instead of being told which chapter I am in.

Contents

CHAPTER 1 Logic is an interface for claims

Rebuild the chapter from first principles

The claim I need to own is this: The model in Logic is an interface for claims controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of logic is an interface for claims. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach logic is an interface for claims on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from the course foundations to proof methods. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

Logic separates the structure of an argument from the subject matter. I rewrite implications and quantifiers before negating them because English intuition is unreliable around words like "all," "some," and "unless."

The easy version

Logic is grammar for truth: it tells how smaller claims combine and what would make the whole claim fail.

A proposition is a statement with a truth value. The implication $P \to Q$ fails only when P is true and Q is false. Its contrapositive $\neg Q \to \neg P$ is equivalent; its converse $Q \to P$ is not.

$$P \to Q \equiv \neg P \vee Q, \qquad \neg(P \wedge Q) \equiv \neg P \vee \neg Q.$$

Quantifiers change the failure condition:

$$\neg(\forall x\, P(x)) \equiv \exists x\, \neg P(x), \qquad \neg(\exists x\, P(x)) \equiv \forall x\, \neg P(x).$$

Common miss

Quantifier order matters. $\forall x \exists y$ allows y to depend on x; $\exists y \forall x$ demands one universal witness.

Solved chapter practice

Practice. Negate: "For every integer n, there exists an integer $m > n$ that is prime."
Solution. Switch each quantifier and negate the predicate: "There exists an integer n such that for every integer $m > n$, m is not prime."

Mastery practice A. Reconstruct logic is an interface for claims from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a logic is an interface for claims problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 2 Proof methods

Rebuild the chapter from first principles

The claim I need to own is this: The model in Proof methods controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of proof methods. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach proof methods on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from logic is an interface for claims to sets, functions, and cardinality. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: a proof method is a way to expose the structure already present in the claim. Direct proof follows the implication, contrapositive reverses it, contradiction denies the conclusion, and counterexample is enough to destroy a universal statement.

The easy version

Choose the route that makes the assumptions useful and the target reachable.

A proof is a finite chain of justified statements from assumptions to conclusion.

- **Direct:** unpack definitions and derive the target.
- **Contrapositive:** prove $\neg Q \to \neg P$ instead of $P \to Q$.
- **Contradiction:** assume the claim false and derive impossibility.
- **Cases:** partition all possibilities and prove each branch.
- **Existence:** construct a witness or prove one must exist.
- **Uniqueness:** assume two candidates and show they coincide.

Worked example

Prove that if n^2 is even, then n is even.
Solution. Prove the contrapositive. If n is odd, $n = 2k + 1$. Then

$$n^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1,$$

which is odd. Therefore n^2 even implies n even.

The move

Start from definitions. If the claim mentions evenness, write $2k$; divisibility, write $a = bd$; subset, take an arbitrary element; injectivity, assume equal outputs. Definitions are executable instructions.

Solved chapter practice

Practice. Prove: if n^2 is even, then n is even.
Solution. Prove the contrapositive. If n is odd, write $n = 2k + 1$. Then $n^2 = 4k^2 + 4k + 1 = 2(2k^2 + 2k) + 1$, which is odd. Therefore if n^2 is even, n cannot be odd and must be even.

Mastery practice A. Reconstruct proof methods from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and

choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a proof methods problem but never states a model or checks the result. Write the shortest useful review of that solution. **Solution.** I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 3 Sets, functions, and cardinality

Rebuild the chapter from first principles

The claim I need to own is this: The model in Sets, functions, and cardinality controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of sets, functions, and cardinality. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I

do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach sets, functions, and cardinality on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from proof methods to induction and invariants. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: sets describe membership; functions describe assignment. Injective means no collision, surjective means no missed target, and bijective means a perfect pairing. Cardinality uses bijections to compare sizes, even for infinite sets.

The easy version

An injective function never reuses an output; a surjective one reaches every output; a bijection does both.

Set identities mirror logic because membership is a proposition. To prove $A = B$, prove both inclusions.

A function $f : A \to B$ is injective if equal outputs force equal inputs, surjective if every $b \in B$ has a preimage, and bijective if both. A function has an inverse exactly when it is bijective.

For finite sets, inclusion-exclusion starts with

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

Infinite sets are compared by bijection. $\mathbb{N}$ and the even natural numbers have the same cardinality via $f(n) = 2n$. A proper subset can have the same size as an infinite set.

Solved chapter practice

Practice. Show that $f : \mathbb{Z} \to \mathbb{Z}$, $f(n) = 2n + 1$, is injective but not surjective.
Solution. If $f(a) = f(b)$, then $2a + 1 = 2b + 1$, so $a = b$; hence injective. But no even integer is in the image, so the function is not surjective onto $\mathbb{Z}$.

Mastery practice A. Reconstruct sets, functions, and cardinality from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a sets, functions, and cardinality problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 4 Induction and invariants

Rebuild the chapter from first principles

The claim I need to own is this: The model in Induction and invariants controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of induction and invariants. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach induction and invariants on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from sets, functions, and cardinality to counting without listing. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: induction is a controlled chain reaction: the base case lights the first step and the inductive step proves one true case forces the next. An invariant is a property that survives every legal move and can rule out impossible states.

The easy version

Prove the first domino falls, then prove any falling domino knocks down the next one.

To prove $P(n)$ for all $n \geq n_0$:

1. prove the base case $P(n_0)$;
2. assume $P(k)$ for an arbitrary $k \geq n_0$;
3. derive $P(k+1)$.

Strong induction assumes all cases through k when proving $k+1$. It is useful when the next object is built from more than the immediately previous one.

Worked example

Prove $1 + 3 + \cdots + (2n-1) = n^2$.
Solution. Base $n = 1$ is immediate. Assume the sum through $2k-1$ is k^2. Adding the next odd number,

$$k^2 + [2(k+1) - 1] = k^2 + 2k + 1 = (k+1)^2.$$

Thus the formula holds for all $n \geq 1$.

Common miss

The induction hypothesis is conditional leverage, not a fact proved for a special k. The step must work for an arbitrary valid k.

Solved chapter practice

Practice. Prove $1 + 3 + \cdots + (2n-1) = n^2$ for $n \geq 1$.
Solution. Base $n = 1$: $1 = 1^2$. Assume the sum through $2k-1$ is k^2. Adding the next odd number gives $k^2 + (2k+1) = (k+1)^2$. Therefore the identity holds for all $n \geq 1$.

Mastery practice A. Reconstruct induction and invariants from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a induction and invariants problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 5 Counting without listing

Rebuild the chapter from first principles

The claim I need to own is this: The model in Counting without listing controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of counting without listing. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach counting without listing on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from induction and invariants to pigeonhole and inclusion-exclusion. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

Counting is about identifying independent choices and correcting overcounting. I ask whether order matters, whether repetition is allowed, and whether objects are distinguishable before selecting a formula.

The easy version

Count the choices step by step, then divide only when different choice histories create the same final object.

If one task has m choices and another independent stage has n, there are mn outcomes. If disjoint cases have m and n outcomes, there are $m + n$.

$$P(n,k) = \frac{n!}{(n-k)!}, \qquad \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Order matters for permutations and does not for combinations.

The binomial theorem is

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k.$$

Its identities often have a counting proof. For example,

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

splits k-person committees by whether a distinguished person is excluded or included.

Solved chapter practice

Practice. How many 5-letter strings from $\{A, B, C, D\}$ contain exactly two A's?
Solution. Choose the two positions for A: $\binom{5}{2} = 10$. Each remaining position has 3 non-A choices, giving $3^3 = 27$. Total $10 \cdot 27 = 270$.

Mastery practice A. Reconstruct counting without listing from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a counting without listing problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 6 Pigeonhole and inclusion-exclusion

Rebuild the chapter from first principles

The claim I need to own is this: The model in Pigeonhole and inclusion-exclusion controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of pigeonhole and inclusion-exclusion. List the known quantities, the unknown, the units or types, and any hidden constraint.

2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.

3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative

derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach pigeonhole and inclusion-exclusion on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from counting without listing to relations and order. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: pigeonhole arguments prove repetition without locating it. Inclusion-exclusion fixes double counting by subtracting overlaps, then adding back higher overlaps that were subtracted too many times.

The easy version

Too many objects in too few boxes forces a shared box; overlapping counts need correction.

If more than n objects enter n boxes, some box receives at least two. In generalized form, placing N objects into k boxes forces one box to contain at least $\lceil N/k \rceil$.

For three finite sets,

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

The signs alternate because each correction overcorrects the next layer.

Worked example

The number of onto functions from an n-element set to a 3-element set is

$$3^n - \binom{3}{1}2^n + \binom{3}{2}1^n,$$

because we subtract functions missing at least one target and restore those missing at least two.

Solved chapter practice

Practice. Among 13 people, prove two were born in the same month.
Solution. The 13 people are pigeons and the 12 months are holes. If each month held at most one birthday, there could be at most 12 people. Since there are 13, at least one month contains two or more birthdays.

Mastery practice A. Reconstruct pigeonhole and inclusion-exclusion from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a pigeonhole and inclusion-exclusion problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 7 Relations and order

Rebuild the chapter from first principles

The claim I need to own is this: The model in Relations and order controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of relations and order. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.

3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach relations and order on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from pigeonhole and inclusion-exclusion to recurrence relations. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: a relation is a chosen set of allowed pairs. Equivalence relations create categories through reflexivity, symmetry, and transitivity; partial orders create hierarchy through reflexivity, antisymmetry, and transitivity.

The easy version

Equivalence says “same kind” and partitions the set; an order says “comes before or is contained in.”

A relation on A is a subset of $A \times A$. An equivalence relation is reflexive, symmetric, and transitive. Its equivalence classes partition A, and every partition defines an equivalence relation.

A partial order is reflexive, antisymmetric, and transitive. Not every pair must be comparable. Hasse diagrams suppress reflexive loops and transitive edges, leaving the cover structure.

Sanity check

Symmetric means aRb implies bRa. Antisymmetric means if both directions hold, then $a = b$. They are not opposites.

Solved chapter practice

Practice. Show congruence modulo 5 is an equivalence relation on $\mathbb{Z}$.
Solution. Reflexive: $5 \mid a - a$. Symmetric: if $5 \mid a - b$, then $5 \mid b - a$. Transitive: if $5 \mid a - b$ and $5 \mid b - c$, then $5 \mid (a - b) + (b - c) = a - c$. Thus it is an equivalence relation.

Mastery practice A. Reconstruct relations and order from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a relations and order problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 8 Recurrence relations

Rebuild the chapter from first principles

The claim I need to own is this: The model in Recurrence relations controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of recurrence relations. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach recurrence relations on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from relations and order to graphs. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: a recurrence defines the present from earlier states. Solving a linear recurrence mirrors differential equations: the characteristic roots produce the basic modes, and initial conditions choose their mixture.

The easy version

The rule tells how to build the next term; a closed form tells the term directly from its index.

A recurrence specifies a sequence from earlier terms. For

$$a_n = ca_{n-1} + da_{n-2},$$

try $a_n = r^n$ and solve $r^2 - cr - d = 0$. Distinct roots give $a_n = Ar_1^n + Br_2^n$; a repeated root gives $(A + Bn)r^n$.

Nonhomogeneous recurrences use a homogeneous solution plus a particular solution, just like linear differential equations.

Worked example

If $a_n = 3a_{n-1} - 2a_{n-2}$, $a_0 = 1$, $a_1 = 4$, roots are $1, 2$. Thus $a_n = A + B2^n$. Conditions give $A = -2$, $B = 3$, so $a_n = 3 \cdot 2^n - 2$.

Solved chapter practice

Practice. Solve $a_n = 3a_{n-1} - 2a_{n-2}$ with $a_0 = 2$, $a_1 = 3$.
Solution. The characteristic equation $r^2 - 3r + 2 = 0$ has roots $1, 2$. Thus $a_n = A + B2^n$. From $A + B = 2$ and $A + 2B = 3$, $B = 1$, $A = 1$. Hence $a_n = 1 + 2^n$.

Mastery practice A. Reconstruct recurrence relations from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a recurrence relations problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 9 Graphs

Rebuild the chapter from first principles

The claim I need to own is this: The model in Graphs controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of graphs. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative

derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach graphs on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from recurrence relations to trees and spanning structure. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

How I actually think about it

A graph keeps only adjacency and ignores geometric drawing details. Degree sums, paths, cycles, connectivity, and bipartiteness let me reason about networks without coordinates.

The easy version

Vertices are objects and edges are relationships; the picture may bend, but who is connected to whom does not change.

A graph $G = (V, E)$ models pairwise connections. A path uses adjacent vertices; a cycle returns to its start; connected means every pair has a path.

The handshake lemma is

$$\boxed{\sum_{v\in V} \deg(v) = 2|E|.}$$

Therefore the number of odd-degree vertices is even.

A connected graph has an Euler circuit exactly when every vertex has even degree, and an Euler trail but not circuit exactly when two vertices have odd degree.

A graph is bipartite exactly when it has no odd cycle. Breadth-first search exposes the two-coloring by distance parity.

Solved chapter practice

Practice. Can a graph have degree sequence $3, 3, 3, 1$?
Solution. The degree sum is 10, which passes the even-sum test. Construct it: connect the three degree-3 vertices to each other, giving each degree 2, then connect the degree-1 vertex to one of them. That makes degrees $3, 2, 2, 1$, not the target. The two remaining vertices cannot gain degree without also increasing another degree. A direct Havel-Hakimi reduction $3, 3, 3, 1 \to 2, 2, 0 \to 1, -1$ fails, so the sequence is not graphical.

Mastery practice A. Reconstruct graphs from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a graphs problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 10 Trees and spanning structure

How I actually think about it

My shorthand for this chapter is: my shorthand for this chapter is: a tree is connected with exactly enough edges to avoid cycles. Removing any edge disconnects it and adding any edge creates one cycle. Those equivalent properties make induction on trees especially natural.

The easy version

A tree is a network with one unique route between every pair of vertices and no spare edge.

A tree is a connected acyclic graph. For a graph with n vertices, any two of these imply the third:

- connected;
- acyclic;
- has $n - 1$ edges.

There is a unique simple path between every pair of vertices. Every nontrivial tree has at least two leaves.

A spanning tree keeps every vertex while removing cycle edges. Depth-first and breadth-first search naturally produce spanning trees; weighted minimum spanning trees are built by safe-edge strategies such as Kruskal or Prim.

The idea

Trees are the minimal connected architecture: remove any edge and connectivity breaks; add any edge and a cycle appears. They sit exactly at the seam between too little and redundant structure.

Rebuild the chapter from first principles

The claim I need to own is this: Trees are the minimal connected architecture: remove any edge and connectivity breaks; add any edge and a cycle appears. They sit exactly at the seam between too little and redundant structure. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of trees and spanning structure. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach trees and spanning structure on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from graphs to cumulative practice. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

Solved chapter practice

Practice. Prove a tree with $n \geq 2$ vertices has at least two leaves.
Solution. Take a longest simple path. If either endpoint had degree greater than 1, it would have a neighbor not already used in the path; otherwise a cycle would appear. Adding that neighbor would extend the path, contradicting maximality. Thus both endpoints have degree 1, so there are at least two leaves.

Mastery practice A. Reconstruct trees and spanning structure from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.
Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a trees and spanning structure problem but never states a model or checks the result. Write the shortest useful review of that solution.
Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 11 Cumulative practice

Rebuild the chapter from first principles

The claim I need to own is this: The model in Cumulative practice controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of cumulative practice. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach cumulative practice on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from trees and spanning structure to written solutions. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

1. Negate: "For every real x, there exists an integer n with $n > x$."
2. Prove that $\sqrt{2}$ is irrational.
3. Prove $(A \cap B)^c = A^c \cup B^c$ by element chasing.
4. Prove by induction that $\sum_{k=1}^{n} k(k+1) = n(n+1)(n+2)/3$.
5. How many length-8 binary strings contain exactly three 1s and do not begin with 1?
6. Show that among 13 people, two have birthdays in the same month.
7. On $\mathbb{Z}$, define aRb when $a - b$ is divisible by 5. Prove this is an equivalence relation and describe classes.
8. Solve $a_n = 4a_{n-1} - 4a_{n-2}$ with $a_0 = 1$, $a_1 = 4$.
9. A connected graph has exactly four odd-degree vertices. Can it have an Euler trail using every edge exactly once? Explain.
10. Prove a tree on n vertices has $n - 1$ edges.

CHAPTER 12 Written solutions

Rebuild the chapter from first principles

The claim I need to own is this: The model in Written solutions controls the work. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of written solutions. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach written solutions on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from cumulative practice to the cumulative course model. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

1. Negation

Move negation through quantifiers and negate the predicate:

$$\exists x \in \mathbb{R}\ \forall n \in \mathbb{Z}, \quad n \leq x.$$

2. Irrationality

Assume $\sqrt{2} = p/q$ in lowest terms. Then $p^2 = 2q^2$, so p^2 and hence p are even: $p = 2k$. Substitution gives $q^2 = 2k^2$, so q is even. Both share factor 2, contradicting lowest terms.

3. De Morgan

For arbitrary x,

$$x \in (A \cap B)^c \iff x \notin A \cap B \iff (x \notin A) \vee (x \notin B) \iff x \in A^c \cup B^c.$$

Thus the sets are equal.

4. Induction

Base $n = 1$: $2 = 1 \cdot 2 \cdot 3/3$. Assume the formula at n. Then

$$\sum_{k=1}^{n+1} k(k+1) = \frac{n(n+1)(n+2)}{3} + (n+1)(n+2) = \frac{(n+1)(n+2)(n+3)}{3}.$$

5. Binary strings

The first bit is fixed as 0. Choose positions for three 1s among the remaining seven: $\binom{7}{3} = 35$.

6. Pigeonhole

People are objects and months are 12 boxes. Thirteen objects in 12 boxes force a shared box.

7. Congruence mod 5

Reflexive: $a - a = 0$ is divisible by 5. Symmetric: if $5|(a-b)$, then $5|(b-a)$. Transitive: divisibility of $a - b$ and $b - c$ implies divisibility of their sum $a - c$. Classes are residues $[0], [1], [2], [3], [4]$ modulo 5.

8. Repeated-root recurrence

Characteristic equation $(r-2)^2 = 0$, so $a_n = (A+Bn)2^n$. From $a_0 = 1$, $A = 1$. From $a_1 = 4$, $2(1+B) = 4$, so $B = 1$. Thus $a_n = (n+1)2^n$.

9. Euler trail

No. A graph with an Euler trail has exactly zero or two odd-degree vertices. Four odd vertices cannot be paired as the start and end of a single trail.

10. Tree edges

Induct on n. A one-vertex tree has zero edges. Every nontrivial tree has a leaf. Remove a leaf and its incident edge; the remaining graph is a tree on $n-1$ vertices, with $n-2$ edges by induction. Restoring the one removed edge gives $n-1$.

Closing frame

Discrete mathematics is the course where correctness becomes a first-class object. Counting arguments, graph structure, and recurrences all depend on precise claims and explicit invariants. The best habit is simple: name the object, state the contract, and make every inference earn the next one.