Discrete Mathematics

10-week lecture notes, worked examples, and exercises (with solutions appendix)

Aligned to sections in Susanna S. Epp, Discrete Mathematics with Applications (5th ed.).

These notes and exercises are original and are intended as a companion to the textbook, not a replacement.

Version: December 2025

How to use these notes

Each week (module) in this document corresponds to the section ranges shown in your course schedule. For each week you'll find: (i) a brief reading guide, (ii) narrative lecture notes, (iii) fully worked examples, and (iv) a problem set.

An appendix at the end contains complete solutions to all of the problem sets. A good workflow is: read the notes → work the examples → attempt the exercises without peeking → then check the appendix.

Notation. This document uses standard symbols: ∈ (is an element of), ⊆ (subset), ∅ (empty set), ℕ (nonnegative integers or positive integers—check your course convention), ℤ, ℚ, ℝ.

Course map (10 weeks)

Week	Textbook sections	Primary theme	Keywords
1	6.1–6.4	Set theory	element method, set identities, power set, Boolean algebra, paradoxes
2	7.1–7.4	Functions + cardinality	domain/codomain, injective/surjective, inverses, composition, countability
3	8.1–8.4	Relations + modular arithmetic	properties, closures, equivalence classes, congruences, inverses mod n
4	9.1–9.4	Counting & probability I	sample spaces, multiplication/addition rules, inclusion–exclusion, pigeonhole
5	9.5–9.7	Counting & probability II	combinations, repetition, binomial coefficients, binomial theorem
6	9.8; 1.4; 4.9	Expected value + intro graphs	probability axioms, expectation, graphs, degree, handshake theorem
7	10.1–10.3	Graph theory I	trails/paths/circuits, Euler, adjacency matrices, isomorphism
8	10.4–10.6	Trees & graph algorithms	trees, rooted trees, spanning trees, shortest paths
9	12.1–12.3	Regular expressions & automata	regex, DFA, minimization, equivalence of states
10	11.1–11.5	Algorithm efficiency	growth rates, O/Ω/Θ, analyzing loops/recurrences

Table of contents

Week 1: Set theory

Reading: Epp §6.1–6.4

Learning objectives

- Use set-builder and roster notation fluently; translate between English statements and sets.
- Prove set identities using the element method (show $x \in \text{LHS} \Leftrightarrow x \in \text{RHS}$).
- Apply common set laws (commutative, associative, distributive, De Morgan) correctly.
- Work with power sets and partitions; recognize when an example disproves a claim.
- Interpret basic Boolean algebra as “algebra of sets” and connect it to logic.

6.1 Sets, subsets, and the element method

A set is a collection of distinct objects called elements. We write $x \in A$ to mean “x is an element of A,” and $x \notin A$ to mean “x is not an element of A.”

A set can be specified by listing its elements (roster notation) or by a defining property (set-builder notation). For example, $\{1, 2, 3\} = \{ x \in \mathbb{Z} : 1 \leq x \leq 3 \}$.

Key definitions

Subset: $A \subseteq B$ means every element of A is also an element of B.

Proper subset: $A \subset B$ means $A \subseteq B$ and $A \neq B$.

Set equality: $A = B$ means $A \subseteq B$ and $B \subseteq A$ (equivalently, $\forall x, x \in A \Leftrightarrow x \in B$).

Common operations: union $A \cup B$, intersection $A \cap B$, difference $A \setminus B$, complement A^c (relative to a universe U).

The most reliable way to prove a set identity is the element method. To prove $A = B$, pick an arbitrary element x and show $x \in A$ iff $x \in B$. To prove $A \subseteq B$, pick arbitrary $x \in A$ and show $x \in B$.

Example 1.1 (Element method: distributive law)

Prove that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

Solution. Let x be arbitrary. Then:

$x \in A \cap (B \cup C) \Leftrightarrow (x \in A)$ and $(x \in B \cup C)$.

The condition $x \in B \cup C$ means $(x \in B)$ or $(x \in C)$. So:

$x \in A \cap (B \cup C) \Leftrightarrow (x \in A)$ and $((x \in B)$ or $(x \in C))$.

Distribute “and” over “or”: this is equivalent to $((x \in A$ and $x \in B)$ or $(x \in A$ and $x \in C))$.

That is, $x \in (A \cap B) \cup (A \cap C)$. Since x was arbitrary, the sets are equal. ■

Two more concepts appear constantly in discrete math:

Power set and partitions

Power set: P(A) is the set of all subsets of A. If A has n elements, then P(A) has 2^n elements.

Partition: A partition of A is a collection of nonempty subsets whose union is A and which are pairwise disjoint.

Example 1.2 (Power set)

Let $A = \{a, b, c\}$. List P(A) and verify that it has $2^3 = 8$ elements.

Solution. The subsets are: $\varnothing$, $\{a\}$, $\{b\}$, $\{c\}$, $\{a,b\}$, $\{a,c\}$, $\{b,c\}$, $\{a,b,c\}$. There are 8, matching 2^3. ■

6.2 Set laws and De Morgan’s laws

Set operations satisfy algebraic laws that look like the laws of arithmetic. These laws let you simplify expressions and prove identities. In practice, you’ll use two proof styles: (1) the element method, and (2) an algebraic proof that rewrites one side into the other using known laws.

De Morgan’s laws (for sets)

$(A \cup B)^c = A^c \cap B^c$

$(A \cap B)^c = A^c \cup B^c$

Example 1.3 (De Morgan via the element method)

Prove $(A \cup B)^c = A^c \cap B^c$.

Solution. Let x be arbitrary.

$x \in (A \cup B)^c \Leftrightarrow x \notin (A \cup B)$.

$x \notin (A \cup B)$ means “x is not in A and not in B,” i.e., $(x \notin A)$ and $(x \notin B)$.

That is equivalent to $(x \in A^c)$ and $(x \in B^c)$, which means $x \in A^c \cap B^c$.

So $x \in (A \cup B)^c \Leftrightarrow x \in A^c \cap B^c$ for arbitrary x, hence the sets are equal. ■

6.3 Disproofs and algebraic proofs

To disprove a universal claim like “for all sets A, B, …, statement S holds,” you only need one counterexample—one choice of sets that makes S false. Good counterexamples are usually small, concrete sets (often subsets of {1,2,3}).

Example 1.4 (Disprove a false set identity)

Claim: $A \setminus (B \cap C) = (A \setminus B) \cap (A \setminus C)$. Is it true?

Solution. Take $A = \{1,2\}$, $B = \{1\}$, $C = \{2\}$.

Compute $B \cap C = \varnothing$, so $A \setminus (B \cap C) = A \setminus \varnothing = \{1,2\}$.

But $A \setminus B = \{2\}$ and $A \setminus C = \{1\}$, so $(A \setminus B) \cap (A \setminus C) = \{2\} \cap \{1\} = \varnothing$.

Left side is {1,2} and right side is ∅, so the claim is false. ■

6.4 Boolean algebra, paradoxes, and why definitions matter

The collection of sets (within a fixed universe U) behaves like a Boolean algebra: ∪ acts like OR, ∩ acts like AND, and complement acts like NOT. This is more than a cute analogy: it lets you translate problems between set identities and logical identities.

Example 1.5 (A Boolean-style simplification)

Simplify the set expression $(A \cap B) \cup (A \cap B^c)$.

Solution. Use distributivity:

$(A \cap B) \cup (A \cap B^c) = A \cap (B \cup B^c)$.

But $B \cup B^c = U$ (the universe), so the expression becomes $A \cap U = A$. ■

Finally, some famous paradoxes show why “set of all sets with property P” must be handled carefully. Russell’s paradox is the classic example: if you allow the set $R = \{ x : x \notin x \}$, then asking whether $R \in R$ creates a contradiction. Modern set theory avoids this by restricting how sets can be formed (axioms rather than unrestricted comprehension).

Week 1 problem set

M1-1. Let $U = \{1,2,3,4,5\}$. Let $A = \{1,2,4\}$, $B = \{2,3,5\}$, $C = \{1,3,4\}$. Compute: (a) $A \cup B$, (b) $A \cap C$, (c) $A \setminus B$, (d) $(B \cup C)^c$.

M1-2. Prove using the element method that $A \cap B \subseteq A$ for all sets A and B.

M1-3. Prove $(A \setminus B) \setminus C = A \setminus (B \cup C)$.

M1-4. True or false? If $A \subseteq B$ then $P(A) \subseteq P(B)$. Prove your answer.

M1-5. How many subsets does a set with 9 elements have? Explain briefly.

M1-6. Let $A = \{1,2,3,4\}$. Consider the collection $\{\{1,2\},\{3\},\{4\}\}$. Is it a partition of A? Justify.

M1-7. Simplify $(A \cup B) \cap (A \cup B^c)$ as much as possible.

M1-8. Let U be a universe and let $X, Y \subseteq U$. Translate the logical statement “$x \in X$ implies $x \in Y$” into an equivalent set containment statement.

Week 2: Functions and cardinality

Reading: Epp §7.1–7.4

Learning objectives

- Use the definition of function precisely (domain, codomain, range) and work with images and preimages.
- Decide whether a function is injective (one-to-one), surjective (onto), or bijective.
- Compute and reason about compositions; know what can and cannot be cancelled in compositions.
- Construct inverse functions when they exist, and prove inverse properties.
- Understand countability: build bijections and use diagonal arguments for uncountability.

7.1 Functions on general sets

A function f from a set A to a set B is a rule that assigns to each $a \in A$ exactly one element $f(a) \in B$. We write $f: A \to B$, call A the domain, B the codomain, and the set $\{ f(a) : a \in A \}$ the range (or image).

Images and preimages

For $S \subseteq A$, the image of S is $f(S) = \{ f(x) : x \in S \} \subseteq B$.

For $T \subseteq B$, the preimage of T is $f^{-1}(T) = \{ x \in A : f(x) \in T \} \subseteq A$.

Note: $f^{-1}(T)$ always makes sense as a set, even when f has no inverse function.

Example 2.1 (Image and preimage)

Let $f: \mathbb{Z} \to \mathbb{Z}$ be defined by $f(n) = n^2 - 1$. Find $f(\{-2,0,3\})$ and $f^{-1}(\{0,3,8\})$.

Solution. Compute the image:

$f(-2) = 4-1 = 3$, $f(0)=-1$, $f(3)=9-1=8$, so $f(\{-2,0,3\}) = \{3, -1, 8\}$.

For the preimage, solve $n^2 - 1 \in \{0,3,8\}$. That means $n^2 \in \{1,4,9\}$.

So $n \in \{\pm1, \pm2, \pm3\}$. Hence $f^{-1}(\{0,3,8\}) = \{-3,-2,-1,1,2,3\}$. ■

7.2 Injective, surjective, bijective; inverse functions

Injective / surjective / bijective

Injective (one-to-one): $f(a_1)=f(a_2) \Rightarrow a_1=a_2$.

Surjective (onto): for every $b \in B$ there exists $a \in A$ with $f(a)=b$.

Bijective: both injective and surjective (equivalently: has an inverse function).

For finite sets, injective and surjective are strongly linked: if $|A|=|B|$ and $f: A\to B$ is injective, then it is automatically surjective (and vice versa). For infinite sets this is not true, so definitions matter.

Example 2.2 (Injective but not surjective)

Define $f: \mathbb{Z} \to \mathbb{Z}$ by $f(n)=2n$. Show f is injective but not surjective.

Solution. If $f(n_1)=f(n_2)$ then $2n_1=2n_2$, so $n_1=n_2$; hence injective.

But f(n) is always even, so there is no n with $f(n)=1$. Therefore f is not onto $\mathbb{Z}$. ■

Example 2.3 (Inverse function on ℝ)

Let $g: \mathbb{R} \to \mathbb{R}$ be $g(x)=x^3$. Find g^{-1} and verify $g(g^{-1}(y))=y$.

Solution. Solve $y=x^3$ for x: $x=\sqrt[3]{y}$. So $g^{-1}(y)=\sqrt[3]{y}$.

Then $g(g^{-1}(y)) = g(\sqrt[3]{y}) = (\sqrt[3]{y})^3 = y$, as required. ■

7.3 Composition of functions

If $f: A\to B$ and $g: B\to C$, the composition $g\circ f: A\to C$ is defined by $(g\circ f)(a)=g(f(a))$. Composition is associative: $h\circ(g\circ f) = (h\circ g)\circ f$ whenever types match. But composition is generally not commutative: $g\circ f$ usually differs from $f\circ g$.

Example 2.4 (Non-commutativity of composition)

Let $f(x)=2x+3$ and $g(x)=x^2$ (both from $\mathbb{R}$ to $\mathbb{R}$). Compute $g\circ f$ and $f\circ g$.

Solution. $(g\circ f)(x)=g(2x+3)=(2x+3)^2=4x^2+12x+9$.

Meanwhile $(f\circ g)(x)=f(x^2)=2x^2+3$.

They are different functions, so composition is not commutative here. ■

A useful fact: if f and g are both bijections, then $(g\circ f)$ is a bijection and $(g\circ f)^{-1} = f^{-1}\circ g^{-1}$. The order reverses, just like for inverses of matrices.

7.4 Cardinality and countability

To compare the “sizes” of sets (including infinite sets), we use bijections. Two sets A and B have the same cardinality if there exists a bijection f: A→B.

Countable vs. uncountable

A set is countably infinite if it has the same cardinality as $\mathbb{N}$ (its elements can be listed as $a_0, a_1, a_2, \ldots$).

A set is countable if it is finite or countably infinite.

A set is uncountable if it is not countable (no listing captures all elements).

Example 2.5 ($\mathbb{Z}$ is countable)

Show that $\mathbb{Z}$ is countably infinite.

Solution. One explicit listing is 0, 1, −1, 2, −2, 3, −3, …

Formally, define $f: \mathbb{N} \to \mathbb{Z}$ by $f(0)=0$ and for $n \geq 1$:

if n is even, $f(n)=n/2$; if n is odd, $f(n)=-(n+1)/2$.

This hits every integer exactly once, so it is a bijection and $\mathbb{Z}$ is countable. ■

Example 2.6 (Diagonal argument idea)

Why is the set of all infinite binary strings uncountable?

Sketch. Suppose, for contradiction, that you could list them as $s_0, s_1, s_2, \ldots$ where each s_i is an infinite 0–1 sequence.

Build a new sequence t by flipping the diagonal bit: set $t(i) = 1$ if $s_i(i)=0$ and $t(i)=0$ if $s_i(i)=1$.

Then t differs from s_i at position i for every i, so t is not equal to any listed sequence—contradiction.

Hence no complete listing exists, so the set is uncountable. ■

Week 2 problem set

M2-1. Let A = {1,2,3,4} and B = {a,b,c}. Define f: A→B by f(1)=a, f(2)=b, f(3)=b, f(4)=c. Find the range of f. Is f injective? Is it surjective?

M2-2. Let $f: \mathbb{R}\to\mathbb{R}$ be $f(x)=x^2$. Find $f(\{-2,-1,0,3\})$. Find $f^{-1}(\{1,4\})$.

M2-3. Prove: If f: A→B and g: B→C are injective, then $g \circ f$ is injective.

M2-4. Prove: If $f: A \to B$ and $g: B \to C$ are surjective, then $g \circ f$ is surjective.

M2-5. Let $f(x)=3x-5$. Find $f^{-1}(x)$. Then compute $(f^{-1} \circ f)(x)$ and $(f \circ f^{-1})(x)$.

M2-6. Give an example of two functions f and g such that $f \circ g$ is the identity on the domain of g, but $g \circ f$ is not the identity on the domain of f.

M2-7. Show that the set $E = \{2n : n \in \mathbb{N}\}$ of even natural numbers is countably infinite by giving a bijection $\mathbb{N} \to E$.

M2-8. Show that the interval $(0,1) \subseteq \mathbb{R}$ is uncountable using a diagonal-style argument with decimal expansions (be careful about $0.4999\ldots = 0.5$ type issues).

Week 3: Relations and modular arithmetic

Reading: Epp §8.1–8.4

Learning objectives

- Represent relations as sets of ordered pairs, directed graphs, and 0–1 matrices.
- Test relations for reflexivity, symmetry, antisymmetry, and transitivity.
- Work with equivalence relations and equivalence classes; move between partitions and equivalence relations.
- Compute and use congruences (mod n), including modular inverses via the Euclidean algorithm.
- Solve basic linear congruences and interpret “mod n” arithmetic in applications (coding/cryptography).

8.1 Relations on sets

A relation R from A to B is any subset of A×B. If (a,b) ∈ R we write aRb. When A=B we say R is a relation on A.

Useful relation operations

Inverse relation: $R^{-1} = \{ (b,a) : (a,b) \in R \}$.

Domain: $\mathrm{dom}(R) = \{ a \in A : \exists b, (a,b) \in R \}$.

Range: $\mathrm{ran}(R) = \{ b \in B : \exists a, (a,b) \in R \}$.

Relations can be represented in three common ways: (1) as a list of ordered pairs, (2) as a directed graph (digraph) on the elements of A, and (3) as a 0–1 matrix M where M[i,j]=1 iff $(a_i,a_j) \in R$.

Example 3.1 (Divisibility as a relation)

Let A = {1,2,3,4}. Define R on A by xRy iff x divides y. List R.

Solution. 1 divides everything, so (1,1),(1,2),(1,3),(1,4) are in R.

2 divides 2 and 4: add (2,2),(2,4).

3 divides 3: add (3,3).

4 divides 4: add (4,4).

So R = {(1,1),(1,2),(1,3),(1,4),(2,2),(2,4),(3,3),(4,4)}. ■

8.2 Reflexivity, symmetry, transitivity

Core properties (for relations on A)

Reflexive: $\forall a \in A$, $(a,a) \in R$.

Symmetric: $(a,b) \in R \Rightarrow (b,a) \in R$.

Antisymmetric: if $(a,b) \in R$ and $(b,a) \in R$ then $a=b$.

Transitive: $(a,b) \in R$ and $(b,c) \in R \Rightarrow (a,c) \in R$.

When a relation fails a property, it helps to exhibit a witness. For example, to show “not symmetric,” find $(a,b) \in R$ but $(b,a) \notin R$.

Example 3.2 (Property check)

For R = “divides” on A = {1,2,3,4}, is R reflexive? symmetric? antisymmetric? transitive?

Solution. Reflexive: yes, because every number divides itself, so $(a,a) \in R$ for all a.

Symmetric: no, since $(1,2) \in R$ but $(2,1) \notin R$.

Antisymmetric: yes. If $a|b$ and $b|a$ (with a,b positive integers), then $a=b$.

Transitive: yes. If $a|b$ and $b|c$ then $a|c$. ■

Sometimes you want the “closest” relation that does have a property. For example, the reflexive closure of R is obtained by adding all missing (a,a) pairs; the symmetric closure adds (b,a) whenever (a,b) is present; and the transitive closure adds all pairs forced by repeated transitivity.

8.3 Equivalence relations and equivalence classes

An equivalence relation is a relation that is reflexive, symmetric, and transitive. Its purpose is to formalize “sameness” under some rule (same remainder mod n, same birthday month, same connected component, …).

Equivalence class

If R is an equivalence relation on A and $a \in A$, the equivalence class of a is $[a] = \{ x \in A : xRa \}$.

Equivalence classes form a partition of A: every element lies in exactly one class.

Example 3.3 (Equivalence classes mod 4)

Define $x \equiv y \pmod 4$ on $\mathbb{Z}$ by “4 divides $x-y$.” List the distinct equivalence classes.

Solution. Two integers are equivalent iff they have the same remainder when divided by 4.

So there are 4 classes: $[0]=\{\dots,-8,-4,0,4,8,\dots\}$, $[1]=\{\dots,-7,-3,1,5,9,\dots\}$, $[2]=\{\dots,-6,-2,2,6,10,\dots\}$, $[3]=\{\dots,-5,-1,3,7,11,\dots\}$. ■

8.4 Modular arithmetic and modular inverses

For integers $n \geq 2$, we write $a \equiv b \pmod n$ when $n \mid (a-b)$. This is an equivalence relation on $\mathbb{Z}$, and arithmetic can be performed “mod n” by working with remainders.

Modular inverse

An integer a has a multiplicative inverse mod n if there exists x such that $ax \equiv 1 \pmod n$.

Such an inverse exists iff $\gcd(a,n)=1$. The extended Euclidean algorithm can find it.

Example 3.4 (Find an inverse mod 26)

Find the inverse of 7 modulo 26.

Solution. We need x such that $7x \equiv 1 \pmod{26}$. Try the Euclidean algorithm:

$26 = 3\cdot7 + 5$, $7 = 1\cdot5 + 2$, $5 = 2\cdot2 + 1$.

Back-substitute: $1 = 5 - 2\cdot2 = 5 - 2\cdot(7 - 1\cdot5) = 3\cdot5 - 2\cdot7$.

And $5 = 26 - 3\cdot7$, so $1 = 3\cdot(26 - 3\cdot7) - 2\cdot7 = 3\cdot26 - 11\cdot7$.

Thus $-11\cdot7 \equiv 1 \pmod{26}$, so $x \equiv -11 \equiv 15 \pmod{26}$.

Check: $7\cdot15=105 \equiv 1 \pmod{26}$. ■

Example 3.5 (Toy RSA workflow)

This example is for arithmetic practice, not security. Let $p=5$ and $q=11$, so $n=pq=55$ and $\varphi(n)=(p-1)(q-1)=40$.

Choose $e=3$ (coprime to 40). Find d such that $ed \equiv 1 \pmod{40}$.

Solution. We need $3d \equiv 1 \pmod{40}$. One solution is $d=27$ because $3\cdot27=81 \equiv 1 \pmod{40}$.

To encrypt a message m (with $0 \le m < n$), compute $c \equiv m^e \pmod{n}$. To decrypt, compute $m \equiv c^d \pmod{n}$.

In practice you would compute powers with fast modular exponentiation rather than expanding huge numbers. ■

Week 3 problem set

M3-1. Let $A=\{1,2,3,4\}$. Define R on A by $(a,b)\in R$ iff $a+b$ is even. List all ordered pairs in R.

M3-2. For the relation R in M3-1, determine whether R is reflexive, symmetric, antisymmetric, and transitive.

M3-3. Let $A=\{a,b,c\}$. Let $R=\{(a,a),(b,b),(c,c),(a,b),(b,a)\}$. Is R an equivalence relation? If yes, list its equivalence classes.

M3-4. Given the partition of $A=\{1,2,3,4,5,6\}$ into blocks $\{1,4\}$, $\{2,5\}$, $\{3,6\}$, write the corresponding equivalence relation R (as a set of ordered pairs).

M3-5. Solve the congruence $9x \equiv 6 \pmod{15}$. List all solutions modulo 15.

M3-6. Find the inverse of 17 modulo 43.

M3-7. Compute (12345 mod 7) and (12345 mod 11) without a calculator by reducing step by step.

M3-8. A message is encrypted with the toy RSA setup $p=5$, $q=11$, $e=3$. If the plaintext is $m=12$, compute the ciphertext $c = m^e \bmod 55$.

Week 4: Counting and probability I

Reading: Epp §9.1–9.4

Learning objectives

- Define sample spaces and events and compute probabilities in equally likely finite models.
- Use the multiplication rule (fundamental counting principle) and possibility trees.
- Use the addition rule and inclusion–exclusion for two sets/events.
- Apply complementary counting to simplify problems.
- Use the pigeonhole principle to prove existence statements.

9.1 Sample spaces, events, and basic probability

A probability model starts with a sample space S: the set of all possible outcomes. An event is a subset $E \subseteq S$. In the simplest "equally likely" setting, each outcome has probability $1/|S|$, so $P(E)=|E|/|S|$.

Example 4.1 (Two heads in three coin flips)

Flip a fair coin 3 times. What is the probability of getting exactly 2 heads?

Solution. The sample space has $2^3=8$ equally likely outcomes.

Exactly two heads occur in HHT, HTH, THH: 3 outcomes.

So P(exactly 2 heads)=3/8. ■

9.2 Possibility trees and the multiplication rule

The multiplication rule says: if a process has k stages, and stage i has n_i choices regardless of earlier choices, then the total number of outcomes is $n_1 \cdot n_2 \cdot \ldots \cdot$n_k. Possibility trees are a diagrammatic way to track stages when the number of choices depends on earlier choices.

Example 4.2 (Counting passwords)

How many 6-character strings can be formed from the alphabet {A,…,Z,0,…,9} if repetition is allowed?

Solution. There are 26+10=36 choices for each position. By the multiplication rule: 36^6 strings. ■

9.3 The addition rule and inclusion–exclusion (two sets)

If A and B are disjoint sets ($A \cap B = \varnothing$), then $|A \cup B| = |A| + |B|$. If they overlap, you must subtract the overlap once: $|A \cup B| = |A| + |B| - |A \cap B|$. The same formula holds for probabilities: $P(A \cup B)=P(A)+P(B)-P(A \cap B)$.

Example 4.3 (Counting divisible numbers 1–100)

How many integers from 1 to 100 are divisible by 2 or 5?

Solution. Let A be multiples of 2, B be multiples of 5.

$|A|=\lfloor 100/2 \rfloor=50$, $|B|=\lfloor 100/5 \rfloor=20$, $|A \cap B|=$multiples of 10: $\lfloor 100/10 \rfloor=10$.

So $|A \cup B|=50+20-10=60$. ■

9.4 The pigeonhole principle

The pigeonhole principle is a small idea with disproportionate power: if you put more objects than boxes, then some box contains at least two objects. The generalized version says: if N objects are placed into k boxes, then some box contains at least $\lceil N/k \rceil$ objects.

Example 4.4 (Birth months)

Show that in any group of 13 people, at least two were born in the same month.

Solution. There are 12 months (boxes) and 13 people (objects). By the pigeonhole principle, some month contains at least two birthdays. ■

Week 4 problem set

M4-1. A fair die is rolled once. What is the probability that the result is (a) even, (b) greater than 4, (c) even or greater than 4?

M4-2. A student must choose a username consisting of 2 letters followed by 3 digits. Letters may repeat; digits may repeat. How many usernames are possible?

M4-3. How many 5-letter strings over {A,B,C} contain at least one A? (Hint: complement.)

M4-4. In a class, 18 students take math, 12 take CS, and 7 take both. How many take at least one of the two courses?

M4-5. Use the pigeonhole principle to show that among any 6 integers, there are two with the same remainder when divided by 5.

M4-6. Show that in any set of 10 distinct integers, there exist two whose difference is divisible by 9.

M4-7. A bag contains 5 red, 4 blue, and 3 green balls. If you draw 2 balls without replacement, what is the probability they are the same color?

M4-8. How many permutations of the letters in the word LEVEL are there? (Treat identical letters as indistinguishable.)

Week 5: Counting and probability II

Reading: Epp §9.5–9.7

Learning objectives

- Use combinations (binomial coefficients) to count subsets and selections without order.
- Solve “stars and bars” problems (combinations with repetition) correctly.
- Apply Pascal’s identity and the binomial theorem.
- Give combinatorial proofs of simple identities.
- Use combinations to compute probabilities (hypergeometric-style counts).

9.5 Combinations and binomial coefficients

When order does not matter, we count combinations. The number of r-element subsets of an n-element set is written “n choose r” and denoted C(n,r) or (n r).

Binomial coefficient

$C(n,r) = n! / (r!(n-r)!)$ for integers $0 \le r \le n$.

Interpretation: number of ways to choose r items from n distinct items.

Example 5.1 (Committee count)

How many 4-person committees can be formed from 10 students?

Solution. Order does not matter, so the answer is $C(10,4)=10!/(4!6!)=210$. ■

A common mistake is to use permutations when combinations are needed. A quick check: if the problem says “committee,” “subset,” “choose,” or “select” (without roles), it’s usually combinations.

9.6 Combinations with repetition (stars and bars)

Sometimes you choose r items allowing repeats from n types. Equivalently, you distribute r identical balls into n distinct boxes (allowing empty boxes).

Stars and bars

Number of solutions in nonnegative integers to $x_1 + x_2 + \dots + x_n = r$ is $C(r+n-1, n-1)$.

Interpretation: number of multisets of size r drawn from n types.

Example 5.2 (Distributing identical items)

How many nonnegative integer solutions are there to $x_1+x_2+x_3 = 10$?

Solution. Here n=3 and r=10, so the count is C(10+3−1, 3−1)=C(12,2)=66. ■

9.7 Pascal’s identity and the binomial theorem

The binomial coefficients satisfy a recursion that matches Pascal’s triangle: C(n,r) = C(n−1,r) + C(n−1,r−1). One way to understand it is combinatorial: when choosing r elements from {1,…,n}, either you choose n or you don’t.

Binomial theorem

$(x + y)^n$ = Σ_{r=0}^{n} C(n,r) x^{n−r} y^{r}.

It expands a power of a sum into a sum of terms weighted by binomial coefficients.

Example 5.3 (Coefficient extraction)

Find the coefficient of x^7 in $(2x - 3)^{10}$.

Solution. Write $(2x - 3)^{10}$ = Σ C(10,r) (2x)^{10−r} (−3)^{r}.

We want exponent 7 on x, so 10−r = 7 ⇒ r=3.

Coefficient = $C(10,3)\cdot 2^7\cdot(-3)^3 = 120\cdot 128\cdot(-27) = -414{,}720$. ■

Week 5 problem set

M5-1. How many ways are there to choose 5 books from a shelf of 12 distinct books?

M5-2. A pizza shop offers 8 toppings. How many distinct 3-topping pizzas are possible (assume no topping is repeated)?

M5-3. How many 6-card hands from a standard 52-card deck contain exactly 2 aces?

M5-4. How many nonnegative integer solutions are there to $x_1+x_2+x_3+x_4 = 15$?

M5-5. How many integer solutions are there to $x_1+x_2+x_3 = 10$ with $x_i \geq 2$ for all i?

M5-6. Use Pascal’s identity to compute C(11,5) from smaller binomial coefficients.

M5-7. Prove the identity C(n,r) = C(n,n−r) combinatorially (in words).

M5-8. Use the binomial theorem to expand $(x+1)^5$ and then evaluate the expansion at x=1 to compute 2^5.

Week 6: Probability axioms, expected value, and intro graphs

Reading: Epp §9.8, §1.4, §4.9

Learning objectives

- Use the probability axioms to derive standard identities (complements, unions, bounds).
- Compute expected values of discrete random variables and use linearity of expectation.
- Model situations with graphs; compute degrees; recognize simple, complete, and bipartite graphs.
- Apply the handshake theorem to relate degrees and edges and to prove impossibility results.

9.8 Probability axioms and expected value

In real applications, “equally likely outcomes” is too restrictive. Instead we define a probability function P on a sample space S that assigns a real number P(E) to each event $E \subseteq S$ and satisfies three axioms:

Probability axioms

1. $0 \leq P(E) \leq 1$ for every event E.

2. $P(S) = 1$.

3. If $E_1, E_2, \ldots$ are pairwise disjoint, then $P(E_1 \cup E_2 \cup \ldots) = P(E_1)+P(E_2)+\ldots$ (countable additivity).

From these axioms you can derive the usual rules such as $P(E^c)=1-P(E)$ and $P(A \cup B)=P(A)+P(B)-P(A \cap B)$.

Expected value

A (discrete) random variable X assigns a number to each outcome.

If X takes values $x_1,\ldots,x\_k$ with probabilities $p_1,\ldots,p\_k$, then $E[X]=\Sigma\, x_i\, p_i$.

Linearity: $E[X+Y]=E[X]+E[Y]$ even when X and Y are dependent.

Example 6.1 (Expected value of a die)

Let X be the result of rolling a fair six-sided die. Compute E[X].

Solution. X takes values 1,2,3,4,5,6 each with probability 1/6.

$E[X] = (1+2+3+4+5+6)/6 = 21/6 = 3.5$. ■

Example 6.2 (Linearity without independence)

Two cards are drawn without replacement from a standard deck. Let X be the number of aces in the two-card hand. Find E[X].

Solution. Let I_1 be the indicator that the first card is an ace, and I_2 the indicator that the second card is an ace.

Then $X = I_1 + I_2$, so $E[X]=E[I_1]+E[I_2]$.

Now $E[I_1]=P(\text{first is ace})=4/52=1/13$. Also $E[I_2]=P(\text{second is ace})=4/52=1/13$ (symmetry).

So $E[X]=2/13$. Note we did not need independence. ■

1.4 Graph basics: the language of graphs

A graph is a mathematical model for “objects connected by links.” A graph G consists of a set V of vertices (nodes) and a set E of edges (connections). Edges may connect two distinct vertices or (in some definitions) a vertex to itself (a loop).

Graph vocabulary (undirected graphs)

Vertices v and w are adjacent if {v,w} is an edge.

The degree deg(v) is the number of incident edges (count a loop twice).

A simple graph has no loops and no multiple edges between the same pair of vertices.

A complete graph K_n connects every pair of distinct vertices.

A bipartite graph has vertices split into two parts with edges only across the split.

4.9 The handshake theorem

The handshake theorem is one of the fastest ways to turn local degree information into global conclusions about a graph.

Handshake theorem

Let G be any finite undirected graph with edge set E. Then $\Sigma_{v\in V} \deg(v) = 2|E|$.

Corollary: The sum of degrees is even.

Corollary: The number of vertices with odd degree is even.

Example 6.3 (Degree sequence sanity check)

Can a simple graph have degree sequence (3,3,3,1)?

Solution. The sum of degrees is $3+3+3+1=10$, which is even, so the handshake theorem does not immediately rule it out.

But in a simple graph on 4 vertices, the maximum possible degree is 3. That part is fine.

Try to realize it: if one vertex has degree 1, it is adjacent to exactly one other vertex. The remaining three vertices must have degrees 3,3,3, which would force each of them to connect to all other vertices—including the degree-1 vertex—giving it degree 3. Contradiction.

So no such simple graph exists. ■

Week 6 problem set

M6-1. Use the probability axioms to prove: if $A \subseteq B$ then $P(A) \leq P(B)$.

M6-2. Let A and B be events with $P(A)=0.6$, $P(B)=0.5$, and $P(A\cap B)=0.2$. Compute $P(A\cup B)$ and $P(A^c \cap B)$.

M6-3. A game pays $10 with probability 0.3 and pays $0 otherwise. What is the expected payout?

M6-4. A fair coin is flipped until the first head appears. Let X be the number of flips. Compute E[X] (hint: use a geometric series or a known formula).

M6-5. Draw a simple graph with 6 vertices whose degrees are (3,3,2,2,2,0), or explain why none exists.

M6-6. A graph has 13 edges and 9 vertices. The degrees of 8 vertices are: 2,2,3,3,3,4,4,4. What is the degree of the 9th vertex?

M6-7. Prove that in any undirected graph, the number of odd-degree vertices is even.

M6-8. How many edges does the complete graph K_8 have? Justify using degrees or combinations.

Week 7: Graph theory I: trails, matrices, isomorphism

Reading: Epp §10.1–10.3

Learning objectives

- Distinguish walks, trails, paths, circuits; recognize Euler trails/circuits and their degree conditions.
- Build adjacency and incidence matrices and use matrix powers to count walks.
- Use graph invariants (degree sequence, components, cycles) to test for non-isomorphism.
- Construct an isomorphism when two graphs are isomorphic (give an explicit vertex mapping).

10.1 Walks, trails, paths, circuits; Euler trails/circuits

A walk is a sequence of vertices where consecutive vertices are joined by edges. A trail is a walk with no repeated edges. A path is a walk with no repeated vertices. A circuit is a walk that starts and ends at the same vertex.

Euler trails and Euler circuits (undirected graphs)

An Euler trail uses every edge exactly once (start and end may differ).

An Euler circuit is an Euler trail that starts and ends at the same vertex.

Criterion (connected graphs, ignoring isolated vertices):

• Euler circuit exists iff every vertex has even degree.

• Euler trail (but not circuit) exists iff exactly two vertices have odd degree.

Example 7.1 (Euler criterion)

A connected graph has degrees (4,2,2,2,2). Does it have an Euler circuit?

Solution. All degrees are even, so an Euler circuit exists. ■

10.2 Matrix representations of graphs

Label the vertices v_1,…,v_n. The adjacency matrix A is the n×n matrix where A[i,j] is the number of edges between v_i and v_j (for a simple graph, 0 or 1). For undirected simple graphs, A is symmetric and has zeros on the diagonal.

Example 7.2 (Adjacency matrix and counting walks)

Let G be the path graph v_1—v_2—v_3. Its adjacency matrix is

A = [[0,1,0],[1,0,1],[0,1,0]].

Compute A^2 and interpret $(A^2)[1,3]$.

Solution. Multiply:

A^2 = [[1,0,1],[0,2,0],[1,0,1]].

The entry $(A^2)[1,3]=1$ counts the number of length-2 walks from v_1 to v_3. Indeed there is exactly one: $v_1 \to v_2 \to v_3$. ■

10.3 Graph isomorphism

Two graphs G and H are isomorphic if they are the same up to relabeling of vertices. Formally, an isomorphism is a bijection $\varphi: V(G) \to V(H)$ such that $\{u,v\}$ is an edge in G iff $\{\varphi(u),\varphi(v)\}$ is an edge in H.

Strategy checklist:

- Compare easy invariants first: number of vertices, number of edges, degree multiset, number of components.
- Compare structural features: triangles, cycles of given lengths, cut vertices, bipartiteness.
- If invariants match, try to build an explicit vertex mapping that preserves adjacency.

Example 7.3 (Non-isomorphism via degrees)

Graph G has degree multiset {3,3,2,2,2}. Graph H has degree multiset {4,2,2,2,1}. Can they be isomorphic?

Solution. No. Isomorphic graphs have the same degree multiset. These differ, so G and H are not isomorphic. ■

Week 7 problem set

M7-1. A connected graph has exactly four vertices of odd degree. Explain why it cannot have an Euler trail.

M7-2. Determine whether the complete bipartite graph $K_{3,3}$ has an Euler circuit, an Euler trail, or neither. Justify using degrees.

M7-3. For the cycle graph C_4 (a square), write the adjacency matrix (label vertices in order around the cycle).

M7-4. Using your adjacency matrix from M7-3, compute A^2 and interpret the diagonal entries.

M7-5. Give two non-isomorphic graphs on 6 vertices that have the same degree multiset (a degree multiset alone is not a complete invariant).

M7-6. Let G be a simple graph with adjacency matrix A. Explain why the entry $(A^3)[i,i]$ counts length-3 closed walks starting at v_i, and how triangles show up in A^3.

M7-7. Decide whether the following degree sequence is graphical (i.e., can occur for a simple graph): (3,3,3,1,1,1). If yes, draw such a graph; if not, justify.

M7-8. Two graphs each have 8 vertices and 10 edges. One has a vertex of degree 6; the other has maximum degree 4. Are they isomorphic? Explain.

Week 8: Trees and graph algorithms

Reading: Epp §10.4–10.6

Learning objectives

- Recognize trees and use equivalent characterizations (connected + acyclic, unique simple path, edges=vertices−1).
- Work with rooted trees (parent/child, levels, height) and m-ary trees.
- Construct spanning trees and minimum spanning trees (MST) for weighted graphs.
- Run a shortest-path algorithm (Dijkstra) on small weighted graphs and interpret results.

10.4 Trees: examples and basic properties

Informally, a tree is a connected graph with no cycles. Trees are the backbone of many data structures and network designs because they are “just enough edges to stay connected.”

Equivalent characterizations of a tree (finite, undirected)

For a graph with n vertices, the following are equivalent:

1) Connected and acyclic.

2) Connected and has exactly n−1 edges.

3) Acyclic and has exactly n−1 edges.

4) There is a unique simple path between any two vertices.

Example 8.1 (Edges in a tree)

A tree has 18 vertices. How many edges does it have?

Solution. Any tree with n vertices has n−1 edges. Here n=18, so edges=17. ■

A common proof pattern: show that adding one edge to a tree creates exactly one cycle, and removing any edge disconnects it. This is the reason spanning trees are “minimally connected.”

10.5 Rooted trees and m-ary trees

A rooted tree is a tree with a distinguished vertex called the root. This induces parent/child relationships: every vertex except the root has a unique parent (the next vertex on the path to the root).

m-ary tree facts

A rooted tree is m-ary if each vertex has at most m children.

A rooted tree is full m-ary if every internal vertex has exactly m children.

In a full m-ary tree with i internal vertices, the number of leaves is $L = (m-1)i + 1$.

Example 8.2 (Leaves in a full binary tree)

A full binary tree ($m=2$) has 10 internal vertices. How many leaves does it have?

Solution. Use $L=(m-1)i+1 = (2-1)\cdot 10+1 = 11$. ■

10.6 Spanning trees and shortest paths

Given a connected graph G, a spanning tree is a subgraph that is a tree and includes all vertices of G. If edges have weights (costs), a minimum spanning tree (MST) is a spanning tree with minimum total weight.

A standard MST method is Kruskal’s algorithm: sort edges by weight and keep adding the next lightest edge that does not create a cycle.

For shortest paths from a source vertex in a graph with nonnegative weights, a standard method is Dijkstra’s algorithm. It repeatedly “locks in” the next vertex with smallest tentative distance.

Pseudocode: Dijkstra’s algorithm

```
Dijkstra(G, source s):
    for each vertex v:
        dist[v] = ∞
        prev[v] = None
    dist[s] = 0
    Q = set of all vertices
    while Q not empty:
        u = vertex in Q with smallest dist[u]
        remove u from Q
        for each neighbor v of u still in Q:
            alt = dist[u] + weight(u,v)
            if alt < dist[v]:
                dist[v] = alt
                prev[v] = u
    return dist, prev
```

Example 8.3 (Dijkstra on a tiny graph)

Suppose edges have weights: s–a (2), s–b (5), a–b (1), a–t (6), b–t (2). Find the shortest distance from s to t.

Solution (sketch). Start with dist[s]=0, others ∞.

Relax from s: dist[a]=2, dist[b]=5. Next pick a (2). Relax from a:

via a→b: dist[b] becomes min(5, 2+1)=3. via a→t: dist[t]=8.

Next pick b (3). Relax from b: dist[t] becomes min(8, 3+2)=5.

So the shortest distance s→t is 5, achieved by s→a→b→t. ■

Week 8 problem set

M8-1. A connected graph has 15 vertices and 14 edges. It has no cycles. Prove it is a tree.

M8-2. A tree has 9 vertices. If you add one new edge between two previously non-adjacent vertices, how many cycles are created? Explain.

M8-3. In a full 3-ary tree, there are 8 internal vertices. How many leaves are there?

M8-4. Draw a rooted tree of height 2 where the root has 3 children, one of those children has 2 children, and all other vertices are leaves. State the number of leaves.

M8-5. Given a connected graph with vertices {1,2,3,4} and edges {12,13,14,23,34}, find a spanning tree (list its edges).

M8-6. Run Kruskal’s algorithm on a graph with vertices {a,b,c,d} and weighted edges: ab(1), ac(5), ad(4), bc(2), bd(6), cd(3). List the MST edges and total weight.

M8-7. Run Dijkstra’s algorithm from source s on the weighted graph: s–a(1), s–b(4), a–b(2), a–t(6), b–t(3). Give the final dist values.

M8-8. Explain why Dijkstra’s algorithm requires nonnegative edge weights by giving a small counterexample with a negative weight.

Week 9: Regular expressions and finite-state automata

Reading: Epp §12.1–12.3

Learning objectives

- Define alphabets, strings, and languages; use Σ* and ε correctly.
- Write and interpret regular expressions using union, concatenation, and Kleene star.
- Design deterministic finite automata (DFA) for simple pattern constraints.
- Simulate a DFA on an input string and decide acceptance.
- Minimize a DFA by identifying equivalent (indistinguishable) states.

12.1 Formal languages and regular expressions

An alphabet Σ is a finite set of symbols. A string over Σ is a finite sequence of symbols from Σ. The empty string is ε. The set of all strings over Σ is Σ*. A language is any subset of Σ*.

Regular expression operators (core ones)

If R and S are regular expressions, then:

• (R | S) denotes union (either a string from R or from S).

• RS denotes concatenation (a string from R followed by a string from S).

• R* denotes Kleene star (zero or more repetitions of strings from R).

Example 9.1 (A simple regex)

Over Σ={a,b}, give a regular expression for all strings that start with a and end with b.

Solution. The middle can be any string over {a,b}, i.e., (a|b)*. So one answer is: a(a|b)*b. ■

Regular expressions are good for describing patterns, but to reason algorithmically we often convert them into automata.

12.2 Deterministic finite-state automata (DFA)

A deterministic finite automaton (DFA) consists of:

DFA definition (informal)

- A finite set of states Q.
- An input alphabet Σ.
- A transition function δ: $Q \times \Sigma \to Q$ (deterministic: exactly one next state).
- A start state $q_0 \in Q$.
- A set of accepting states $F \subseteq Q$.

A string is accepted if, starting at q_0 and following transitions symbol by symbol, the final state is in F.

Example 9.2 (DFA for “even number of 1s” in binary strings)

Design a DFA over Σ={0,1} that accepts exactly those strings with an even number of 1s.

Solution. Use two states: E (even so far) and O (odd so far). Start at E. On input 0, stay in the same state; on input 1, toggle between E and O. Accepting state is E. ■

Transition table for Example 9.2

```
State | on 0 | on 1
------+------|------
  E   |  E   |  O
  O   |  O   |  E
```

12.3 Simplifying (minimizing) finite-state automata

Different DFAs can recognize the same language. Minimization produces an equivalent DFA with as few states as possible. The key idea is state equivalence: two states p and q are equivalent if no future input can distinguish them (they lead to acceptance or rejection together for every continuation).

A practical minimization method is partition refinement: start by splitting states into accepting vs non-accepting, then repeatedly split blocks when transitions on some symbol go to different blocks.

Example 9.3 (Mini minimization example)

Suppose a DFA has states {A,B,C,D}, accepting states {C,D}. If A and B both go to C on input 0 and to D on input 1, and C and D both loop to themselves on both symbols, then A and B are equivalent and can be merged.

Reason. Starting from A or B, after one symbol you land in C or D in the same way. From then on, C and D behave deterministically. No string can separate A from B, so they are indistinguishable. ■

Week 9 problem set

M9-1. Over Σ={a,b}, write a regular expression for all strings that contain the substring “ab” at least once.

M9-2. Describe in words the language of the regular expression (a|b)*aa(a|b)* over Σ={a,b}.

M9-3. Construct a DFA over Σ={0,1} that accepts strings that end with 01.

M9-4. For your DFA in M9-3, trace the computation on the input strings 01, 101, 1110, and 1001. Indicate accept/reject.

M9-5. Design a DFA over Σ={0,1} that accepts binary strings representing numbers divisible by 3 (allow leading zeros).

M9-6. Give a regular expression over Σ={0,1} for all strings of length at least 2 whose first and last symbols are the same.

M9-7. Minimize the DFA with states {S,A,B}, alphabet {0,1}, start state S, accepting states {A}, and transitions: δ(S,0)=A, δ(S,1)=B, δ(A,0)=A, δ(A,1)=B, δ(B,0)=A, δ(B,1)=B. (Which states can be merged?)

M9-8. Explain why the language L = { a^n b^n : n ≥ 0 } is not regular (give a pigeonhole/pumping-lemma style argument, at a high level).

Week 10: Analysis of algorithm efficiency

Reading: Epp §11.1–11.5

Learning objectives

- Compare growth rates of common functions (logarithmic, polynomial, exponential).
- Use big-O, big-Ω, and big-Θ definitions and prove simple bounds.
- Analyze the worst-case step counts of basic algorithms (loops, nested loops).
- Analyze divide-and-conquer algorithms (binary search, merge sort) at the level of growth rates.
- Solve simple recurrences or recognize common forms (e.g., $T(n)=2T(n/2)+n$).

11.1 Growth of functions: intuition and graphs

When we analyze algorithms, we usually care about how running time grows with input size n. At large n, exact constants matter less than growth rate. Common families (in increasing growth order) include $\log n$, n, $n \log n$, n^2, n^3, 2^n, $n!$.

Example 10.1 (Which grows faster?)

Which function grows faster as $n\to\infty$: $n \log_2 n$ or n^{1.5}?

Answer (reason). n^{1.5} grows faster. One way to see it: divide by n log n to compare:

n^{1.5} / (n log n) = n^{0.5} / log n → ∞ because √n eventually dominates log n. ■

11.2 Big-O, big-Ω, and big-Θ

Asymptotic notation (core definitions)

$f(n)$ is $O(g(n))$ if $\exists C,n_0$ such that for all $n \geq n_0$, $0 \leq f(n) \leq C\cdot g(n)$.

$f(n)$ is $\Omega(g(n))$ if $\exists c,n_0$ such that for all $n \geq n_0$, $0 \leq c\cdot g(n) \leq f(n)$.

$f(n)$ is $\Theta(g(n))$ if $f(n)$ is both $O(g(n))$ and $\Omega(g(n))$.

Think: O is an upper bound (“grows no faster than”), Ω is a lower bound (“grows at least as fast as”), and Θ is a tight bound (“same order of growth”).

Example 10.2 (Prove a Θ bound)

Show that $f(n)=3n^2+5n+1$ is $\Theta(n^2)$.

Solution. For $n \geq 1$, we have $3n^2 \leq 3n^2+5n+1 \leq 3n^2+5n^2+n^2 = 9n^2$.

So with $c=3$, $C=9$, and $n_0=1$, we get $3n^2 \leq f(n) \leq 9n^2$ for all $n \geq 1$.

Hence $f(n)$ is $\Theta(n^2)$. ■

11.3 Application: analyzing simple loops

A practical approach: decide what a “basic operation” is (comparison, addition, array access), then count how many times it executes as a function of n. You can then convert that exact count into an asymptotic class.

Counting example: nested loop

```
# Example: nested loop
count = 0
for i = 1 to n:
    for j = 1 to n:
        count = count + 1
```

The inner statement runs n times for each of n values of i, so total iterations = $n \cdot n = n^2$. This is $\Theta(n^2)$.

11.4 Exponential and logarithmic functions

Logarithms turn multiplication into addition: $\log(ab)=\log a + \log b$. This is why algorithms that repeatedly halve a problem size often run in logarithmic time. Exponential functions like 2^n grow extremely fast and usually indicate brute-force behavior.

Example 10.3 (Binary search running time)

Binary search halves the search interval each step. After k steps, the remaining size is about n/2^k.

We stop when n/2^k ≈ 1, meaning 2^k ≈ n, so $k \approx \log_2 n$.

Thus binary search runs in $\Theta(\log n)$ comparisons. ■

11.5 Application: divide-and-conquer and recurrences

Divide-and-conquer algorithms often lead to recurrences. A classic example is merge sort:

Merge sort recurrence (informal)

To sort n items, merge sort sorts two halves of size n/2 and then merges them in linear time.

This gives $T(n) = 2T(n/2) + cn$, which solves to $T(n) = \Theta(n \log n)$.

Example 10.4 (Solving T(n)=2T(n/2)+n by expansion)

Assume n is a power of 2 and $T(1)=1$. Expand:

$T(n)=2T(n/2)+n$

$=2[2T(n/4)+n/2]+n = 4T(n/4)+2n$

$=8T(n/8)+3n$

After k steps: $T(n)=2^k T(n/2^k)+k\cdot n$.

Stop when $n/2^k=1 \Rightarrow k=\log_2 n$. Then:

$T(n)=n\cdot T(1)+(\log_2 n)\cdot n = n + n \log_2 n = \Theta(n \log n)$. ■

Week 10 problem set

M10-1. Order the following functions from slowest-growing to fastest-growing (as $n\to\infty$): $\log n$, n, $n \log n$, n^2, 2^n.

M10-2. Show that $7n+20$ is $\Theta(n)$.

M10-3. Show that n^2 is $O(n^3)$ and that n^3 is not $O(n^2)$.

M10-4. A loop runs i from 1 to n, and inside it runs j from 1 to i. How many times does the inner statement execute? Give a Θ bound.

M10-5. In the worst case, sequential search of an array of length n checks every element. Give the exact number of comparisons and its Θ class.

M10-6. Binary search on a sorted array of length n makes at most $\lceil \log_2(n+1) \rceil$ comparisons (up to constants). Explain why this is $O(\log n)$.

M10-7. Solve the recurrence $T(n)=T(n/2)+n$ with $T(1)=1$ for n a power of 2. Give Θ classification.

M10-8. Which grows faster: $n \log n$ or $10n$? At approximately what n does $n \log_2 n$ exceed $10n$? (A rough estimate is fine.)

Appendix: Solutions to problem sets

The solutions below correspond to the labeled exercises in Weeks 1–10. If you find yourself reading a solution and thinking “I see it,” stop and re-derive the key step on your own—that’s where most of the learning happens.

Week 1 solutions

M1-1. Compute each set step by step (universe $U=\{1,2,3,4,5\}$).

(a) $A \cup B = \{1,2,4\} \cup \{2,3,5\} = \{1,2,3,4,5\}$.

(b) $A \cap C = \{1,2,4\} \cap \{1,3,4\} = \{1,4\}$.

(c) $A \setminus B = \{$ elements of A not in B $\} = \{1,4\}$.

(d) First $B \cup C = \{2,3,5\} \cup \{1,3,4\} = \{1,2,3,4,5\}=U$, so $(B \cup C)^c = \varnothing$.

M1-2. Let x be arbitrary and assume $x \in A \cap B$. By definition of intersection, $x \in A$ and $x \in B$. In particular, $x \in A$. Therefore every element of $A \cap B$ is an element of A, so $A \cap B \subseteq A$.

M1-3. We prove set equality by the element method. Let x be arbitrary.

$x \in (A \setminus B) \setminus C \Leftrightarrow x \in (A \setminus B)$ and $x \notin C \Leftrightarrow (x \in A$ and $x \notin B)$ and $x \notin C$.

This is equivalent to $x \in A$ and $(x \notin B$ and $x \notin C)$, i.e., $x \in A$ and $x \notin (B \cup C)$.

Thus $x \in (A \setminus B) \setminus C \Leftrightarrow x \in A \setminus (B \cup C)$. Since x was arbitrary, the sets are equal.

M1-4. True. Suppose $A \subseteq B$. Let $S \in P(A)$. By definition of power set, $S \subseteq A$. Since $A \subseteq B$, transitivity of $\subseteq$ gives $S \subseteq B$. Hence $S \in P(B)$.

So every element of P(A) is an element of P(B), i.e., $P(A) \subseteq P(B)$.

M1-5. A set with n elements has 2^n subsets (each element is either “in” or “out”). With $n=9$, that is $2^9 = 512$ subsets.

M1-6. Yes. Each block is nonempty; the blocks are pairwise disjoint; and their union is $\{1,2\}\cup\{3\}\cup\{4\} = \{1,2,3,4\}$. Therefore it is a partition of A.

M1-7. Use the distributive identity $(X \cup Y) \cap (X \cup Z) = X \cup (Y \cap Z)$. With $X=A$, $Y=B$, $Z=B^c$:

$(A \cup B) \cap (A \cup B^c) = A \cup (B \cap B^c) = A \cup \varnothing = A$.

M1-8. The statement “for all x, if $x \in X$ then $x \in Y$” means every element of X is also an element of Y. That is exactly the subset relation $X \subseteq Y$.

Week 2 solutions

M2-1. The range is the set of outputs: $\{f(1),f(2),f(3),f(4)\} = \{a,b,b,c\} = \{a,b,c\}$.

Not injective because $f(2)=b=f(3)$ but $2\neq3$.

Surjective because every element of $B=\{a,b,c\}$ appears as an output.

M2-2. Compute the image: $f(\{-2,-1,0,3\}) = \{ (-2)^2, (-1)^2, 0^2, 3^2 \} = \{4,1,0,9\}$.

For the preimage, solve $x^2 \in \{1,4\}$. That happens when $x=\pm1$ or $x=\pm2$. So $f^{-1}(\{1,4\}) = \{-2,-1,1,2\}$.

M2-3. Let f: A→B and g: B→C be injective. To show g∘f is injective, assume $(g\circ f)(a_1)=(g\circ f)(a_2)$.

Then $g(f(a_1)) = g(f(a_2))$. Since g is injective, $f(a_1)=f(a_2)$. Since f is injective, $a_1=a_2$. Hence g∘f is injective.

M2-4. Let f: A→B and g: B→C be surjective. We must show g∘f is surjective onto C.

Take any c ∈ C. Because g is onto, there exists b ∈ B with g(b)=c. Because f is onto, there exists a ∈ A with f(a)=b.

Then (g∘f)(a) = g(f(a)) = g(b) = c. Hence g∘f is surjective.

M2-5. Solve $y = 3x - 5$ for x: $x = (y+5)/3$. So $f^{-1}(x) = (x+5)/3$.

Now $(f^{-1}\circ f)(x) = f^{-1}(3x-5) = ((3x-5)+5)/3 = x$.

And $(f\circ f^{-1})(x) = f((x+5)/3) = 3\cdot(x+5)/3 - 5 = x$.

M2-6. Here is one concrete example. Let A={1} and B={1,2}. Define g: A→B by g(1)=1. Define f: B→A by f(1)=1 and f(2)=1.

Then (f∘g)(1)=f(g(1))=f(1)=1, so f∘g is the identity on A.

But g∘f is not the identity on B, because $(g\circ f)(2)=g(f(2))=g(1)=1 \neq 2$.

M2-7. A bijection ℕ→E is f(n)=2n (assuming ℕ={0,1,2,…}).

It is one-to-one because $2n_1=2n_2 \Rightarrow n_1=n_2$, and onto because every even number has the form 2n.

M2-8. Assume, for contradiction, that (0,1) is countable. Then we can list its elements: x_1, x_2, x_3, … with decimal expansions.

Construct a new number y in (0,1) by choosing its digits so that the n-th digit of y differs from the n-th digit of x_n (e.g., change 1→2 and 2→1, and avoid using 9).

Then y differs from x_n in the n-th decimal place for every n, so y is not equal to any x_n. This contradicts that the list contained all numbers in (0,1).

Therefore (0,1) is uncountable.

Week 3 solutions

M3-1. The sum a+b is even exactly when a and b have the same parity (both odd or both even).

Odd elements are 1 and 3, giving pairs (1,1),(1,3),(3,1),(3,3).

Even elements are 2 and 4, giving pairs (2,2),(2,4),(4,2),(4,4).

So R has 8 ordered pairs total.

M3-2. Reflexive: yes, because a+a is always even, so (a,a)∈R for every a.

Symmetric: yes, because if $a+b$ is even then $b+a$ is even.

Antisymmetric: no, since $(1,3)$ and $(3,1)$ are both in R but $1 \neq 3$.

Transitive: yes. If $a+b$ and $b+c$ are even, then a and b have the same parity and b and c have the same parity, so a and c have the same parity, hence $a+c$ is even and $(a,c) \in R$.

M3-3. R is reflexive because $(a,a),(b,b),(c,c)$ are included. It is symmetric because (a,b) and (b,a) are both included, and all (x,x) are symmetric.

For transitivity, the only nontrivial checks involve a and b: since aRb and bRa, we also have aRa and bRb, which are in R. Anything involving c only relates c to itself.

So R is an equivalence relation. The equivalence classes are $\{a,b\}$ and $\{c\}$.

M3-4. The equivalence relation consists of all pairs within each block:

$\{(1,1),(1,4),(4,1),(4,4)\} \cup \{(2,2),(2,5),(5,2),(5,5)\} \cup \{(3,3),(3,6),(6,3),(6,6)\}$.

M3-5. Solve $9x \equiv 6 \pmod{15}$. Compute $\gcd(9,15)=3$ and 3 divides 6, so solutions exist.

Divide the congruence by 3: $3x \equiv 2 \pmod{5}$.

The inverse of 3 mod 5 is 2 (since $3 \cdot 2=6 \equiv 1$ mod5). Multiply both sides by 2:

$x \equiv 4 \pmod{5}$.

Thus modulo 15, the solutions are $x \in \{4, 9, 14\}$.

M3-6. Use the Euclidean algorithm: $43=2 \cdot 17+9$, $17=1 \cdot 9+8$, $9=1 \cdot 8+1$, so $\gcd(17,43)=1$.

Back-substitute: $1=9-1 \cdot 8 = 9-(17-1 \cdot 9)=2 \cdot 9-17 = 2 \cdot (43-2 \cdot 17)-17 = 2 \cdot 43-5 \cdot 17$.

So $-5 \cdot 17 \equiv 1 \pmod{43}$, meaning $17^{-1} \equiv -5 \equiv 38 \pmod{43}$.

M3-7. Reduce by division:

$12345 = 7 \cdot 1763 + 4$, so $12345 \bmod 7 = 4$.

$12345 = 11 \cdot 1122 + 3$, so $12345 \bmod 11 = 3$.

M3-8. Compute $c = m^e \bmod 55$ with $m=12$ and $e=3$:

$12^3 = 1728$. Divide by 55: $55 \cdot 31=1705$, remainder 23.

So the ciphertext is $c = 23$.

Week 4 solutions

M4-1. Sample space size is 6 (outcomes 1–6).

(a) Even outcomes: $\{2,4,6\}$ so $P=3/6=1/2$.

(b) Greater than 4: $\{5,6\}$ so $P=2/6=1/3$.

(c) Even or greater than 4: $\{2,4,5,6\}$ so $P=4/6=2/3$.

M4-2. Two letters: $26 \cdot 26$ choices. Three digits: $10 \cdot 10 \cdot 10$ choices. Multiply:

Total usernames $= 26^2 \cdot 10^3 = 676 \cdot 1000 = 676{,}000$.

M4-3. Total 5-letter strings over {A,B,C}: $3^5=243$.

Complement: strings with no A use only {B,C}, so $2^5=32$.

Therefore strings with at least one A: $243-32=211$.

M4-4. Use inclusion–exclusion: $|\text{Math} \cup \text{CS}| = |\text{Math}|+|\text{CS}|-|\text{Both}| = 18+12-7 = 23$.

M4-5. There are 5 possible remainders modulo 5 (0,1,2,3,4) but 6 integers. By the pigeonhole principle, two integers share a remainder mod 5.

M4-6. There are 9 possible remainders modulo 9, but you have 10 distinct integers. Two must share the same remainder mod 9, and their difference is divisible by 9.

M4-7. Total ways to choose 2 balls: $C(12,2)=66$.

Favorable ways (same color): $C(5,2)+C(4,2)+C(3,2)=10+6+3=19$.

So the probability is 19/66.

M4-8. LEVEL has 5 letters with L repeated twice and E repeated twice.

Distinct permutations = $5!/(2!\cdot 2!) = 120/4 = 30$.

Week 5 solutions

M5-1. Choosing 5 distinct books from 12 without order gives $C(12,5) = 792$.

M5-2. A 3-topping pizza corresponds to choosing 3 toppings out of 8, order irrelevant: $C(8,3)=56$.

M5-3. Choose 2 of the 4 aces and 4 of the 48 non-aces:

Number of hands = $C(4,2)\cdot C(48,4) = 6\cdot 194{,}580 = 1{,}167{,}480$.

M5-4. Count nonnegative solutions to $x_1+x_2+x_3+x_4=15$ using stars and bars:

$C(15+4-1,4-1)=C(18,3)=816$.

M5-5. Let $y_i = x_i-2$, so $y_i \geq 0$ and $y_1+y_2+y_3 = 10-6 = 4$.

Number of solutions = $C(4+3-1,3-1)=C(6,2)=15$.

M5-6. Pascal’s identity: $C(11,5)=C(10,5)+C(10,4)=252+210=462$.

M5-7. Combinatorial proof: $C(n,r)$ counts r-element subsets of an n-element set.

Choosing r elements to include is equivalent to choosing which $n-r$ elements to exclude, giving $C(n,n-r)$.

Therefore $C(n,r)=C(n,n-r)$.

M5-8. $(x+1)^5$ = Σ_{r=0}^5 C(5,r) x^{5−r} = $x^5+5x^4+10x^3+10x^2+5x+1$.

Plug in x=1: $1+5+10+10+5+1 = 32 = 2^5$.

Week 6 solutions

M6-1. Assume $A \subseteq B$. Then B can be written as a disjoint union $B = A \cup (B \setminus A)$.

By additivity on disjoint events, $P(B)=P(A)+P(B \setminus A) \geq P(A)$. Hence $P(A) \leq P(B)$.

M6-2. Use inclusion–exclusion: $P(A \cup B)=P(A)+P(B)-P(A \cap B)=0.6+0.5-0.2=0.9$.

Also $A^c \cap B$ is the part of B outside A, so $P(A^c \cap B)=P(B)-P(A \cap B)=0.5-0.2=0.3$.

M6-3. Let X be the payout. Then $E[X]=10 \cdot 0.3 + 0 \cdot 0.7 = 3$. So the expected payout is $3.

M6-4. Let X be the number of flips until the first head. Condition on the first flip:

With probability 1/2 you get a head immediately ($X=1$). With probability 1/2 you get a tail and then you “start over,” so $X=1+X'$ where X' has the same distribution as X.

So $E[X] = (1/2) \cdot 1 + (1/2) \cdot (1 + E[X])$.

Solve: $E[X] = 1/2 + 1/2 + (1/2)E[X] \Rightarrow (1/2)E[X]=1 \Rightarrow E[X]=2$.

M6-5. One construction with vertices $v_1,\ldots,v_6$ (v_6 isolated):

Edges among $v_1,\ldots,v_5$: v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_2v_5, v_4v_5.

Degrees are: $\deg(v_1)=3$, $\deg(v_2)=3$, $\deg(v_3)=2$, $\deg(v_4)=2$, $\deg(v_5)=2$, $\deg(v_6)=0$, as required.

M6-6. By the handshake theorem, the sum of all degrees is $2|E| = 2 \cdot 13 = 26$.

The given 8 degrees sum to $2+2+3+3+3+4+4+4 = 25$.

So the 9th degree must be $26-25 = 1$.

M6-7. Let the degrees be $d_1,\ldots,d\_n$. The handshake theorem gives $d_1+\ldots+d\_n = 2|E|$, which is even.

A sum of integers is even iff there are an even number of odd addends. Therefore the number of odd-degree vertices is even.

M6-8. In K_8 every vertex has degree 7, so the sum of degrees is $8 \cdot 7=56$.

By the handshake theorem, $56 = 2|E|$, so $|E|=28$.

(Equivalently: choose 2 endpoints out of 8: $C(8,2)=28$ edges.)

Week 7 solutions

M7-1. An Euler trail exists in a connected graph iff exactly 0 or 2 vertices have odd degree.

If the graph has 4 odd-degree vertices, it fails this criterion, so it cannot have an Euler trail.

M7-2. In $K_{3,3}$ each vertex connects to all 3 vertices in the opposite part, so every vertex has degree 3 (odd).

There are 6 odd-degree vertices. An Euler circuit requires all degrees even, and an Euler trail requires exactly two odd degrees.

Therefore $K_{3,3}$ has neither an Euler circuit nor an Euler trail.

M7-3. Label the cycle vertices 1–2–3–4–1. The adjacency matrix is:

A = [[0,1,0,1],[1,0,1,0],[0,1,0,1],[1,0,1,0]].

M7-4. For the matrix A in M7-3, multiply $A^2 = A\cdot A$ to get:

A^2 = [[2,0,2,0],[0,2,0,2],[2,0,2,0],[0,2,0,2]].

The diagonal entry $(A^2)[i,i]=2$ counts length-2 walks that start and end at vertex i: go to either neighbor and return.

M7-5. Example pair on 6 vertices with the same degree multiset but not isomorphic:

- G = cycle C_6 (one 6-cycle). It is connected and every vertex has degree 2.
- H = disjoint union of two triangles $K_3 \cup K_3$. It has two components and every vertex has degree 2.

Both have degree multiset {2,2,2,2,2,2} but they are not isomorphic because one is connected and the other is not.

M7-6. The entry (A^k)[i,j] counts the number of length-k walks from v_i to v_j (this follows from how matrix multiplication sums over intermediate steps).

So $(A^3)[i,i]$ counts length-3 closed walks starting at v_i.

A triangle contributes closed walks of length 3: from each of its 3 vertices you can traverse the triangle in 2 directions, giving 6 closed walks total per triangle.

Therefore the total number of triangles can be extracted from $\text{trace}(A^3)/6$ in a simple graph.

M7-7. Yes, it is graphical. One explicit construction: take vertices A,B,C as a triangle (edges AB,BC,CA) so each has degree 2, then connect A to D, B to E, and C to F.

Degrees become: deg(A)=3, deg(B)=3, deg(C)=3, deg(D)=deg(E)=deg(F)=1, which matches (3,3,3,1,1,1).

M7-8. No. The maximum degree is an isomorphism invariant. One graph has a vertex of degree 6 while the other has maximum degree 4, so they cannot be isomorphic.

Week 8 solutions

M8-1. A tree is defined as a connected graph with no cycles. The graph is given to be connected and acyclic, so it is a tree.

(The extra fact “15 vertices and 14 edges” is consistent with the tree property $|E|=|V|-1$.)

M8-2. Exactly one cycle is created. In a tree there is a unique simple path between any two vertices u and v.

Adding a new edge {u,v} closes that unique path into a single cycle, and no other cycle can appear because all other edges are unchanged.

M8-3. For a full m-ary tree with i internal vertices, the number of leaves is $L=(m-1)i+1$.

Here m=3 and i=8, so $L=(3-1)\cdot 8+1=16+1=17$ leaves.

M8-4. One valid shape: root r has children A,B,C. Vertex A has children A_1 and A_2. Vertices B,C,A_1,A_2 are leaves.

Number of leaves = 4.

M8-5. One spanning tree is the star at 1: edges {12,13,14}.

It contains all 4 vertices, has 3 edges, and has no cycle, so it is a spanning tree.

M8-6. Sort edges by weight: ab(1), bc(2), cd(3), ad(4), ac(5), bd(6).

Kruskal picks ab, then bc (no cycle), then cd (no cycle). Now all 4 vertices are connected with 3 edges, so we stop.

MST edges: {ab, bc, cd}. Total weight = 1+2+3 = 6.

M8-7. Run Dijkstra from s. Initialize dist[s]=0, dist[a]=1, dist[b]=4, dist[t]=∞.

Pick a next (dist 1). Relax: b can be improved via a: 1+2=3 so dist[b]=3. t via a: dist[t]=7.

Pick b next (dist 3). Relax t via b: 3+3=6 so dist[t]=6.

Final distances: dist[s]=0, dist[a]=1, dist[b]=3, dist[t]=6.

M8-8. Counterexample: vertices s,a,b with edges s→a (1), s→b (2), b→a (−2).

The true shortest distance from s to a is 0 via s→b→a. But Dijkstra would finalize a at distance 1 before considering the negative edge, so it can fail.

This is why Dijkstra requires all edge weights to be nonnegative.

Week 9 solutions

M9-1. A standard answer is (a|b)*ab(a|b)*. It says: any prefix, then “ab”, then any suffix.

M9-2. (a|b)*aa(a|b)* describes all strings over {a,b} that contain “aa” as a contiguous substring at least once.

M9-3. Use states that remember whether the most recent symbol was 0 and whether the last two symbols were 01.

States: q0 (start / last symbol not 0), q1 (last symbol is 0), q2 (last two symbols are 01). Accepting state: q2.

Transitions: δ(q0,0)=q1, δ(q0,1)=q0; δ(q1,0)=q1, δ(q1,1)=q2; δ(q2,0)=q1, δ(q2,1)=q0.

M9-4. Trace using the DFA from M9-3:

01: q0→q1→q2 (accept).

101: q0→q0→q1→q2 (accept).

1110: q0→q0→q0→q0→q1 (reject).

1001: q0→q0→q1→q1→q2 (accept).

M9-5. Use 3 states for remainders mod 3: r0 (accept), r1, r2. Reading a bit b updates the remainder by new = (2·old + b) mod 3.

Transitions:

r0: on 0→r0, on 1→r1;

r1: on 0→r2, on 1→r0;

r2: on 0→r1, on 1→r2.

M9-6. Strings of length at least 2 whose first and last symbols match are either:

- start and end with 0: 0(0|1)*0, or
- start and end with 1: 1(0|1)*1.

So one regex is: 0(0|1)*0 | 1(0|1)*1.

M9-7. Start with partition {A} (accepting) and {S,B} (non-accepting).

Check S and B: on 0 both go to A; on 1 both go to B, which is in the same non-accepting block.

So S and B are equivalent and can be merged. The minimized DFA has 2 states: one accepting state A and one non-accepting state [S,B].

M9-8. High-level pumping argument: suppose a DFA with k states recognized L={a^n b^n}. Consider the k+1 strings a, a^2, …, a^{k+1}.

As you read these prefixes, by pigeonhole two different lengths a^i and a^j (i < j) lead to the same state.

From that state, the DFA cannot “remember” how many a’s were read, so it would accept both a^i b^i and a^j b^i (or reject both), contradicting that only equal counts should be accepted.

Therefore L is not regular.

Week 10 solutions

M10-1. From slowest to fastest growth as n→∞: log n, n, n log n, n^2, 2^n.

M10-2. For n ≥ 1, 7n ≤ 7n+20 ≤ 7n+20n = 27n.

So with c=7, C=27, and n_0=1, we have c·n ≤ 7n+20 ≤ C·n for all n ≥ n_0, hence 7n+20 is Θ(n).

M10-3. For n ≥ 1, $n^2 \le n^3$, so n^2 is O(n^3) (take C=1, n_0=1).

To see n^3 is not O(n^2), suppose $n^3 \le Cn^2$ for all n ≥ n_0. Dividing by n^2 gives n ≤ C for all large n, impossible. So n^3 is not O(n^2).

M10-4. The inner statement runs 1+2+…+n times = n(n+1)/2. Therefore it is Θ(n^2).

M10-5. Worst-case sequential search checks every element once, so it makes exactly n comparisons.

Thus the running time is Θ(n).

M10-6. Ceilings do not change asymptotic class, so it suffices to bound $\log_2(n+1)$.

For $n \geq 1$, $n+1 \leq 2n$, so $\log_2(n+1) \leq \log_2(2n) = 1 + \log_2 n$.

Hence $\lceil \log_2(n+1) \rceil$ is $O(\log n)$.

M10-7. Assume n=2^k. Expand:

$T(n)=T(n/2)+n = T(n/4)+n/2+n = \ldots = T(1) + (n + n/2 + n/4 + \ldots + 2)$.

The geometric sum is $2n-2$, so $T(n)=1+(2n-2)=2n-1$. Therefore $T(n)=\Theta(n)$.

M10-8. $n \log n$ grows faster than $10n$ because $\log n \to \infty$.

Using base-2 logs: $n \log_2 n > 10n$ when $\log_2 n > 10$, i.e., $n > 2^{10} = 1024$.

So around $n \approx 1024$, $n \log_2 n$ starts exceeding $10n$ (roughly).