Problem 1: Relation Properties

For each relation below, determine whether it is *reflexive, irreflexive, coreflexive, symmetric, antisymmetric, asymmetric, transitive, left/right Euclidean, connected, semi-connected*. Organize your findings in a table and provide counterexamples for properties that don't hold.

1. **Proximity relation:** For real numbers, define $x \ R \ y$ iff $|x - y| \leq 1$.
2. **Subset hierarchy:** For all subsets of $\{a, b, c\}$, define $\langle A, B \rangle \in R$ iff $A \subseteq B$.
3. **Communication flow:** For users $\{a, b, c, d\}$, relation R has adjacency matrix:

$$[\![R]\!] = \texttt{[([0], [1], [0], [1]) ([0], [0], [0], [1]) ([1], [1], [0], [0]) ([0], [0], [1], [0])]}$$

4. **Game dominance:** In rock-paper-scissors, define $x \ R \ y$ iff "x beats y".
5. **Modulo equivalence:** For natural numbers, define $x \ R \ y$ iff $x \equiv y \pmod 7$.

Problem 2: Social Network Relations

Consider a social network with users $U = \{\text{Alice}, \text{Bob}, \text{Carol}, \text{Dave}, \text{Eve}\}$ and three types of connections:

- **Friendship** F: mutual (symmetric) friend relations
- **Following** L: who follows whom (for updates)
- **Trust** T: who trusts whom (for recommendations)

Given these specific relations:

- $F = \{\langle \text{Alice}, \text{Bob} \rangle, \langle \text{Bob}, \text{Alice} \rangle, \langle \text{Carol}, \text{Dave} \rangle, \langle \text{Dave}, \text{Carol} \rangle\}$
- $L = \{\langle \text{Alice}, \text{Carol} \rangle, \langle \text{Bob}, \text{Dave} \rangle, \langle \text{Carol}, \text{Alice} \rangle, \langle \text{Dave}, \text{Eve} \rangle, \langle \text{Eve}, \text{Bob} \rangle\}$
- $T = \{\langle \text{Alice}, \text{Bob} \rangle, \langle \text{Bob}, \text{Carol} \rangle, \langle \text{Carol}, \text{Dave} \rangle, \langle \text{Dave}, \text{Alice} \rangle\}$

Part (a): Relation Operations

1. Find the *mutual connections* $B = F \cup (L \cap L^{-1})$. What does $L \cap L^{-1}$ represent?
2. Compute the *influence chain* $I = L \circ T$.
3. Compute the *trust chain* $J = T \circ L$.
4. Compare I and J. What do they represent?
5. Is there exists an *influencer* — a user to whom all other users have a direct edge in $L \cup I$?
6. Is there exists a *trust hub* — a user to whom all other users have a direct edge in $T \cup J$?

Part (b): Property Preservation

1. **Transitivity:** Is it true that if R_1 and R_2 are transitive relations, then $R_1 \cap R_2$ is also transitive? Either prove this statement or construct a counterexample on $\{1, 2, 3\}$.
2. **Equivalence:** Prove that if R and S are equivalence relations on the same set, then $R \cap S$ is also an equivalence relation.
3. **Symmetry:** Can the union of two symmetric relations not be symmetric?

Part (c): Trust Propagation

Trust can spread in chains: you might trust someone because a person you trust recommends them. Let $T^2 = T \circ T$ represent *second-hand trust* (trust over 2 people), $T^3 = T^2 \circ T$ represent *third-degree trust* (trust over 3 steps), and so on. Define the *k-fold trust network* as $T^{[k]} = \bigcup_{i=1}^{k} T^i$.

1. Compute $T^{[2]}$ and $T^{[3]}$.
2. When does the trust network *stabilize*? Find the smallest k where $T^{[k+1]} = T^{[k]}$.
3. Show that the *ultimate trust network* $T^+ = \bigcup_{n=1}^{\infty} T^n$ (containing all possible trust connections) also satisfies the transitivity property, for arbitrary initial trust relation T.

Problem 3: Equivalence Relations

Part (a): Equinumerosity Relation

The *equinumerosity relation* $\approx$ is defined as: $A \approx B$ iff $|A| = |B|$.

1. Prove that $\approx$ is an equivalence relation over finite sets.
2. Prove that $\approx$ is an equivalence relation over infinite sets.[1]
3. Find the quotient set of $\mathcal{P}(\{a, b, c, d\})$ by $\approx$.

Part (b): Modular Arithmetic

Consider the relation R_m on integers: $a \; R_m \; b$ iff $a \equiv b \pmod{m}$ for fixed $m \geq 1$.

1. Prove that R_m is an equivalence relation for any $m \geq 1$.
2. Describe the equivalence classes of R_7 and find the quotient set $\mathbb{Z}/_{R_7}$.
3. Show that the quotient set $\mathbb{Z}/_{R_m}$ has exactly m elements.

Part (c): String Permutation

Define relation $\sim$ on the set of all finite strings over alphabet $\Sigma = \{a, b, c\}$ where $s_1 \sim s_2$ iff s_2 can be obtained from s_1 by rearranging (permuting) its characters.

For example, abc $\sim$ bca and aab $\sim$ aba, but abc $\nsim$ ab and abc $\nsim$ abb.

1. Prove that $\sim$ is an equivalence relation.
2. Find all equivalence classes for strings over Σ of length 3.
3. How many equivalence classes exist for Σ-strings of length n?

Problem 4: Similarity Networks and Tolerance Relations

Academic researchers often collaborate when they share common expertise areas. We model this collaboration potential using the *θ-similarity relation* R_θ, where $\theta \in [0, 1] \subseteq \mathbb{R}$ is a similarity threshold parameter. Two researchers A and B are *θ-similar* (denoted $A \; R_\theta \; B$) if their Jaccard similarity coefficient satisfies:

$$\mathcal{J}(A, B) = \frac{|A \cap B|}{|A \cup B|} \geq \theta$$

where A and B represent the finite sets[2] of expertise areas for each researcher.

[1] For infinite sets, $|A| = |B|$ means there exists a bijection between A and B.
[2] Either consider only non-empty sets, or define $\mathcal{J}(\emptyset, \emptyset) = 1$.

Consider six researchers with expertise in the following areas:

Researcher	Expertise Areas
赖 (Lài)	Graph Theory, High-Performance Computing, Bioinformatics, Databases
石 (Shí)	Internet of Things, Cryptography, Formal Methods, Embedded Systems
邱 (Qiū)	Cryptography, Algorithms, Bioinformatics
魏 (Wèi)	High-Performance Computing, Databases, Graph Theory, Algorithms
辛 (Xīn)	Embedded Systems, Algorithms, Databases
朱 (Zhū)	Formal Methods, Internet of Things, Cryptography, Databases

Part (a): Building the Collaboration Network

For a similarity threshold of $\theta = 0.25$:

1. Calculate all pairwise Jaccard similarities $\mathcal{J}(A, B)$ between the six researchers.
2. Determine which pairs are 0.25-similar.
3. Draw the collaboration network graph $G_{0.25}$, where vertices are researchers and edges connect 0.25-similar pairs.

Part (b): Properties of the Similarity Relation

1. Prove that the θ-similarity relation R_θ is a *tolerance relation* for any $\theta \in [0; 1]$ by showing it is reflexive and symmetric.
2. For the specific researcher data, determine whether the relation $R_{0.25}$ is transitive. If not, provide a counterexample where transitivity fails.
3. For which value(s) of θ (if any) does R_θ become an equivalence relation for arbitrary sets?

Part (c): Network Structure and Dynamics

1. Identify all research *clusters* (the connected components) in the graph $G_{0.25}$.
2. As θ increases from 0 to 1, the collaboration network R_θ loses edges. Find all *critical* values of θ at which the number of connected components in the network changes.
3. What is the maximum value of θ for which the collaboration network of our six researchers remains connected (*i.e.*, is a single connected component)?

Part (d): Researcher Impact Analysis

1. **Diversity analysis:** Define the *diversity index* $d(X)$ of a researcher X as their number of expertise areas, $d(X) = |X|$. For each researcher, compute their diversity index and their average Jaccard similarity with all others:

$$\overline{\mathcal{J}}(X) = \frac{1}{5} \sum_{Y \neq X} \mathcal{J}(X, Y)$$

   Does the researcher with the highest diversity index also have the highest average similarity?

2. **Network resilience:** If one researcher were to leave the network, whose absence would cause the most fragmentation? Determine which researcher's removal from the $\theta = 0.25$ network results in the maximum number of connected components.

Problem 5: Boolean Matrix and Transitive Closure

Part (a): Boolean Matrix Product

Any relation $R \subseteq M^2$ on a finite set M with $n = |M|$ can be represented as an $n \times n$ Boolean matrix $[\![R]\!] = [r_{ij}]$, where $r_{ij} = 1$ iff $\langle m_i, m_j \rangle \in R$, and 0 otherwise.

The *Boolean product* of two matrices $A \odot B = [c_{ij}]$ is defined as: $c_{ij} = \bigvee_k (a_{ik} \wedge b_{kj})$.

1. Compute the Boolean product:

$$[([1], [0], [1]) \quad ([0], [1], [0]) \quad ([1], [1], [0])] \odot [([0], [1], [1]) \quad ([1], [0], [1]) \quad ([0], [1], [0]$$

2. For the Boolean matrix M below, compute its Boolean square $M \odot M$ and cube $M \odot M \odot M$. Interpret these results in terms of paths in the corresponding directed graph.

$$M = [([1], [1], [0]) \quad ([0], [1], [1]) \quad ([1], [0], [1])]$$

3. Prove that if R and S are relations, then the matrix of $S \circ R$ equals $[\![R]\!] \odot [\![S]\!]$.

Part (b): Warshall's Algorithm for Transitive Closure

The *Warshall algorithm* computes the transitive closure R^+ of a relation R on a set M of size n. Given the $n \times n$ adjacency matrix A of the relation, the algorithm iteratively computes a sequence of matrices $A^{(0)}, A^{(1)}, \ldots, A^{(n)}$.

It starts with $A^{(0)} = A$ and uses the following recurrence for $k = 1, \ldots, n$:

$$A_{ij}^{(k)} = A_{ij}^{(k-1)} \vee \left(A_{ik}^{(k-1)} \wedge A_{kj}^{(k-1)} \right)$$

The final matrix $A^{(n)}$ is the adjacency matrix of the transitive closure R^+.

1. Apply Warshall's algorithm to compute the transitive closure of R on $M = \{1, 2, 3, 4\}$:

$$R = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 1 \rangle, \langle 3, 4 \rangle\}$$

   Show the matrix at each iteration step, $A^{(0)}, A^{(1)}, A^{(2)}, A^{(3)}$, and the final result $A^{(4)}$.
2. Prove by induction on k that $A_{ij}^{(k)} = 1$ if and only if there is a path from vertex i to j using only intermediate vertices from the set $\{1, \ldots, k\}$.
3. Compare the computational complexity of Warshall's algorithm with the naive approach of computing $A^+ = A \vee A^2 \vee \ldots \vee A^n$ using Boolean matrix products (powers).

Problem 6: Proof Validation and Counterexamples

Part (a): Find the Error

Critique the following "proof":

"Theorem": If relation R is symmetric and transitive, then R is reflexive.

"Proof": Let $a \in A$. Take $b \in A$ such that $\langle a, b\rangle \in R$. Since R is symmetric, $\langle b, a\rangle \in R$. By transitivity, with $\langle a, b\rangle \in R$ and $\langle b, a\rangle \in R$, we get $\langle a, a\rangle \in R$.

1. Identify the logical error.
2. Provide a counterexample showing a symmetric and transitive relation that isn't reflexive.

Part (b): Closure Operations

Find a relation R on $\{a, b, c\}$ such that the symmetric closure of the reflexive closure of the transitive closure of R is *not* transitive.

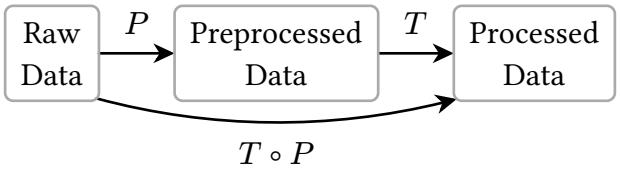
Hint: Work backwards — start with a non-transitive relation and see what R could produce it.

Problem 7: Data Pipeline Composition

Modern data processing systems rely heavily on chaining operations together. Understanding how mathematical properties are preserved (or lost) through the composition of steps is crucial for building reliable pipelines.

Part (a): Reversible Data Transformations

Consider a multi-stage data processing pipeline where data flows through different formats:



Let P represent the preprocessing step (*e.g.*, converting human-filled Excel to machine-readable JSON) and T represent the transformation step (*e.g.*, JSON to analytics results). The complete pipeline is the *composition* $T \circ P$, which directly transforms raw data to final results. In data engineering, it's often crucial to trace data provenance backwards — if we see a result, we want to know which original data produced it.

For relations $R \subseteq A \times B$ and $S \subseteq B \times C$, prove that the inverse of the composition equals the composition of individual inverse steps: $(S \circ R)^{-1} = R^{-1} \circ S^{-1}$.

This composition property ensures that data provenance tracking works correctly in complex pipelines.

Part (b): Pipeline Integrity Analysis

In real-world data systems, we often need to ensure that certain properties are preserved through multi-stage processing. Consider a concrete pipeline:

- $R =$ set of raw customer records (with potential duplicates and missing fields)
- $C =$ set of cleaned customer records (deduplicated, standardized)
- $F =$ set of customer feature vectors (for machine learning)

Let $f : R \to C$ be the data cleaning function and $g : C \to F$ be the feature extraction function. The complete pipeline is their *function composition* $g \circ f : R \to F$, and we need to analyze how properties are preserved through composition:

1. **Uniqueness preservation:** If cleaning never merges distinct customers (injective f) and feature extraction never merges distinct records (injective g), does the complete pipeline preserve customer uniqueness (injective $g \circ f$)?
2. **Coverage guarantee:** If cleaning produces all possible clean record types (surjective f) and extraction produces all possible feature vectors (surjective g), does the pipeline produce all possible features (surjective $g \circ f$)?
3. **Pipeline diagnostics:** If the complete pipeline preserves customer uniqueness, can we conclude the cleaning step never merges customers?
4. **Error isolation:** If the complete pipeline preserves customer uniqueness, can we conclude the feature extraction never merges records?
5. **Intermediate coverage analysis:** If the complete pipeline produces all possible feature vectors, can we conclude the cleaning step produces all possible clean record types?
6. **Output coverage analysis:** If the complete pipeline produces all possible feature vectors, can we conclude the extraction step produces all possible features from clean data?

For each property, either prove the statement or provide a concrete counterexample using realistic data processing scenarios (*e.g.*, filtering operations, aggregations, joins).

Problem 8: Cardinality and Infinity

For each set below, determine whether it is *countable* (same size as natural numbers) or *uncountable* (strictly larger than natural numbers). Provide clear justifications, including explicit bijections or diagonalization arguments where appropriate.

1. The set of rational[3] numbers $\mathbb{Q}$.
2. The power set of natural numbers $\mathcal{P}(\mathbb{N})$.
3. The set of all functions $f : \mathbb{N} \to \mathbb{N}$.
4. The union of countably many countable sets.
5. The set of all computer programs in a particular programming language.
6. The set of real roots of all quadratic equations $ax^2 + bx + c = 0$ with integer coefficients.

[3] A rational number is a fraction m/n, where $m \in \mathbb{Z}$ is an integer and $n \in \mathbb{N}^+$ is a natural number.

Problem 9: Partial Orders

Part (a): Divisibility Poset

Consider $H = \{1, 2, 4, 5, 10, 12, 20\}$ with *divisibility*[4] relation $x \ R \ y$ iff $x \mid y$.

Define grading function $\rho(n)$ to be the sum of exponents in prime factorization of n. For example: $\rho(20) = \rho(2^2 \cdot 5^1) = 2 + 1 = 3$.

1. Verify that R is a partial order.
2. Is R a total order? Explain.
3. Draw the Hasse diagram for $\langle H, R \rangle$ with elements arranged by grade $\rho(n)$.
4. Find all minimal, maximal, minimum, and maximum elements, if they exist.
5. Perform a topological sort of H.

Part (b): Harmonic Ordering

In music theory, notes can be ordered based on harmonic principles. We'll explore two such orderings on the 12-note chromatic scale: $N = \{\mathrm{C}, \mathrm{C\sharp}, \mathrm{D}, \mathrm{D\sharp}, \mathrm{E}, \mathrm{F}, \mathrm{F\sharp}, \mathrm{G}, \mathrm{G\sharp}, \mathrm{A}, \mathrm{A\sharp}, \mathrm{B}\}$.

1. **Circle of Fifths Precedence:** The circle of fifths defines a precedence relation $\preceq$ on the notes:[5]

$$\mathrm{C} \preceq \mathrm{G} \preceq \mathrm{D} \preceq \mathrm{A} \preceq \mathrm{E} \preceq \mathrm{B} \preceq \mathrm{F\sharp} \preceq \mathrm{C\sharp} \preceq \mathrm{G\sharp} \preceq \mathrm{D\sharp} \preceq \mathrm{A\sharp} \preceq \mathrm{F}$$

   Verify that this relation is a partial order and draw its Hasse diagram.

2. **Consonance Dominance:** Notes can also be ordered by their consonance relative to a root note (here, C). The hierarchy of consonance levels is given by the table below.

Level	Category	Notes	Interval
0	Perfect Unison	C	Unison
1	Perfect Consonances	G, F	Perfect 5th, 4th
2	Imperfect Consonances	E, A	Major 3rd, 6th
3	Near Consonances	D, B	Major 2nd, 7th
4	Mild Dissonances	C♯, D♯, G♯, A♯	Minor intervals
5	Maximum Dissonance	F♯	Tritone

   Define the *consonance dominance* relation $\lhd$ such that $x \lhd y$ if note x is strictly more consonant than note y, that is, $\text{level}(x) < \text{level}(y)$. A note at a certain level dominates all notes at higher-numbered (less consonant) levels.

   - Is $\lhd$ reflexive? Antisymmetric? Transitive? Is it a partial order?
   - Draw the Hasse diagram for the poset $\langle N, \lhd \rangle$.
   - Identify all maximal and minimal elements in this poset.

[4] A number x *divides* y (denoted $x \mid y$) if there exists an integer k such that $y = k \cdot x$.

[5] For this problem, we treat this as a linear sequence and ignore the wrap-around from F back to C.

Problem 10: Advanced Topics

Part (a): Well-Founded and Well-Ordered Relations

A poset is *well-founded* if every non-empty subset has a *minimal* element. A poset is *well-ordered* if it is a well-founded total order (or, equivalently, if every non-empty subset has a *least* element).

1. Is lexicographic order well-founded on the set of *finite* lowercase English strings?
2. Is lexicographic order well-founded on the set of *infinite* lowercase English strings?
3. Construct an example of a poset that is well-founded but not well-ordered.

Part (b): Partition Refinement Lattices

A partition α of a set S is a *refinement* of a partition β, denoted[6] $\alpha \trianglelefteq \beta$, if every block of α is a subset of some block of β.

Example: For the set $S = \{a, b, c\}$, the partition $\alpha = \{\{a\}, \{b\}, \{c\}\}$ is a refinement of $\beta = \{\{a, b\}, \{c\}\}$, because the blocks $\{a\}$ and $\{b\}$ are subsets of $\{a, b\}$, and $\{c\}$ is a subset of $\{c\}$.

1. Prove that the set of all partitions of a set S, ordered by the refinement relation $\trianglelefteq$, forms a lattice.
2. For $S = \{a, b, c\}$, list all 5 possible partitions[7] and draw the Hasse diagram of the partition lattice.
3. Draw the Hasse diagram of the partition lattice for the set $S = \{a, b, c, d\}$.
4. Compare the structure of the partition refinement lattice to the Boolean lattice (the poset of subsets of S ordered by $\subseteq$).
5. Is partition refinement lattice distributive?
6. Is partition refinement lattice complemented?

Submission Guidelines:
- Organize solutions clearly with problem numbers and parts.
- For proofs: state assumptions, show logical steps, conclude with QED.
- For disproof: provide specific counterexamples with verification.
- For computational problems: show work and double-check answers.
- Use tables, diagrams, and visual aids where helpful.

Grading Rubric:
- Mathematical accuracy and completeness: 50%
- Proof quality and logical reasoning: 30%
- Clarity, organization, and presentation: 20%

[6] We say that "α is *finer* than β" or "β is *coarser* than α".

[7] The number of partitions of an n-element set is the Bell number B_n. For example, $B_3 = 5$.