

Learning breakdown

Dưới đây là bảng tổng hợp các thông tin từ tài liệu bạn cung cấp, được tổ chức theo từng môn học. Mỗi bảng bao gồm **Khái niệm cốt lõi**, **Bài toán trọng tâm**, **Câu hỏi lớn**, **Ứng dụng trong TCS (Lý thuyết Khoa học Máy tính)** và **Bài toán kinh điển** (bao gồm cả các bài toán từ phần cuối tài liệu).

1. Đại số trừu tượng (Abstract Algebra)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Nhóm: Cấu trúc với phép toán kết hợp, phần tử đơn vị, nghịch đảo (VD: số nguyên với phép cộng).- Vành: Cấu trúc với cộng và nhân (VD: số nguyên).- Trường: Vành mà mọi phần tử $\neq 0$ có nghịch đảo nhân (VD: số thực, số phức).- Đồng cấu: Hàm bảo toàn cấu trúc.- Lý thuyết Galois: Liên kết trường mở rộng và nhóm đối xứng.
Bài toán trọng tâm	<ul style="list-style-type: none">- Phân loại nhóm, vành, trường dựa trên tính chất.- Tìm đồng cấu, đẳng cấu giữa các cấu trúc.- Giải bài toán đối xứng (VD: nhóm hoán vị, đối xứng đa giác).- Ứng dụng trong mật mã (VD: RSA).
Câu hỏi lớn	<ul style="list-style-type: none">- Làm sao mô tả và phân loại cấu trúc đại số tổng quát?- Cấu trúc đại số giải quyết bài toán tính toán (mã hóa, tối ưu) thế nào?- Ánh xạ giữa cấu trúc đại số giúp hiểu hệ thống phức tạp ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Mật mã học: Nhóm, trường hữu hạn (RSA).- Lý thuyết mã hóa: Vành, trường cho mã sửa lỗi.- Thiết kế thuật toán: Nhóm trong thuật toán đồ thị, tối ưu hóa.
Bài toán kinh điển	<ul style="list-style-type: none">- Phân loại nhóm hữu hạn: Liệt kê nhóm hữu hạn (cyclic, Abel, không Abel).- Tìm đồng cấu nhóm: Tìm đồng cấu từ G sang H.- Bài toán từ đẳng cấu: Xác định từ trong nhóm có là phần tử đơn vị không.- Nhóm con Sylow: Tìm nhóm con Sylow của nhóm hữu hạn.- Phân loại nhóm Abel: Phân loại nhóm Abel hữu hạn thành tổng trực tiếp nhóm cyclic.- Bài toán Galois: Nghiên cứu tính giải được của đa thức qua nhóm đối xứng.

2. Logic

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Logic mệnh đề: Mệnh đề đúng/sai, phép nối (AND, OR, NOT).- Logic vị từ: Mở rộng với biến, lượng từ (\forall, \exists).- Lý thuyết mô hình: Quan hệ công thức logic và cấu trúc toán học.- Lý thuyết chứng minh: Phân tích tính đúng đắn chứng minh.- Tính toán logic: Logic trong hệ thống tính toán (temporal, description logic).
Bài toán trọng tâm	<ul style="list-style-type: none">- Xây dựng hệ thống logic để mô tả, kiểm chứng hệ thống tính toán.- Chứng minh/bác bỏ công thức logic.- Kiểm tra tính thỏa mãn (SAT).- Thiết kế công cụ tự động hóa suy luận (trình giải SAT, kiểm chứng mô hình).
Câu hỏi lớn	<ul style="list-style-type: none">- Biểu diễn tri thức, lập luận chính xác trong máy tính thế nào?- Công thức logic có thể chứng minh/bác bỏ không? (Vấn đề quyết định)- Tự động hóa suy luận logic trong AI, kiểm chứng phần mềm ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Kiểm chứng hình thức: Kiểm tra phần cứng/phần mềm.- AI: Suy luận, biểu diễn tri thức.- Ngữ nghĩa ngôn ngữ lập trình: Định nghĩa ý nghĩa chương trình.
Bài toán kinh điển	<ul style="list-style-type: none">- Thỏa mãn (SAT): Tìm gán giá trị cho công thức CNF để đúng.- Kiểm chứng mô hình: Xác định hệ thống thỏa mãn công thức logic.- Quyết định logic vị từ: Kiểm tra công thức đúng mọi mô hình.- Gödel: Hệ tiên đề đủ mạnh có mệnh đề không chứng minh được.- Thompson: Liên quan đến tính quyết định của hệ thống logic.

3. Toán rời rạc (Discrete Mathematics)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Quan hệ, hàm: Liên hệ giữa tập hợp.- Đồ thị, cây: Mô hình hóa mạng, hệ thống.- Tổ hợp đếm: Đếm cách sắp xếp/lựa chọn.- Lý thuyết tập hợp rời rạc: Phép toán trên tập hữu hạn.- Thuật toán cơ bản: Phân tích, thiết kế thuật toán.
Bài toán trọng tâm	<ul style="list-style-type: none">- Phân tích, thiết kế thuật toán trên đồ thị, cây.- Đếm số cấu hình/tổ hợp (VD: số cách xếp n vật).- Tối ưu hóa trên cấu trúc rời rạc (đường đi ngắn nhất, luồng tối đa).- Mô hình hóa hệ thống tính toán.
Câu hỏi lớn	<ul style="list-style-type: none">- Mô hình hóa, giải bài toán thực tế bằng cấu trúc rời rạc thế nào?- Tối ưu hóa thuật toán trên cấu trúc rời rạc đến đâu?- Đếm/liệt kê tất cả cấu hình khả thi trong hệ thống phức tạp ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Thiết kế thuật toán: Phân tích độ phức tạp, hiệu suất.- Mạng máy tính: Đồ thị mô hình hóa kết nối.- Tối ưu hóa: Tổ hợp trong lập lịch, phân bổ tài nguyên.
Bài toán kinh điển	<ul style="list-style-type: none">- Đường đi ngắn nhất: Tìm đường đi ngắn nhất trong đồ thị có trọng số.- Cây bao trùm tối thiểu: Tìm cây với tổng trọng số nhỏ nhất.- Luồng tối đa: Tìm luồng lớn nhất trong mạng.- Tô màu bản đồ: Tô bản đồ với tối đa 4 màu.- Hanoi Tower: Di chuyển đĩa theo quy tắc.- Sắp xếp topo: Sắp xếp đỉnh trong DAG theo thứ tự topological.

4. Tổ hợp (Combinatorics)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Đếm: Tính số cách chọn, sắp xếp (hoán vị, tổ hợp).- Bao hàm-loại trừ: Đếm tập hợp phức tạp.- Hàm sinh: Giải bài toán đếm, phân tích chuỗi.- Tổ hợp đồ thị: Tính chất tổ hợp của đồ thị.- Thiết kế tổ hợp: Khối, ma trận.
Bài toán trọng tâm	<ul style="list-style-type: none">- Đếm số cách sắp xếp/chọn với ràng buộc.- Phân tích cấu trúc tổ hợp trong đồ thị (số vòng, cây).- Thiết kế cấu trúc tổ hợp (mã hóa, lập lịch).- Giải bài toán tối ưu tổ hợp (ghép đôi, bao phủ).
Câu hỏi lớn	<ul style="list-style-type: none">- Có bao nhiêu cách thực hiện nhiệm vụ/cấu hình hệ thống?- Tối ưu hóa cấu trúc tổ hợp trong bài toán thực tế thế nào?- Tổng quát hóa, phân loại cấu trúc tổ hợp ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Phân tích thuật toán: Tính số phép toán cần thiết.- Mật mã học: Thiết kế mã, giao thức.- Tối ưu hóa: Lập kế hoạch, phân bổ tài nguyên.
Bài toán kinh điển	<ul style="list-style-type: none">- Bao phủ tập hợp: Tìm tập con nhỏ nhất bao phủ tập gốc.- Đếm hoán vị: Tính số cách sắp xếp với ràng buộc.- Thiết kế khối: Xây dựng khối thỏa mãn ràng buộc.- Ramsey: Tìm số lớn nhất để đồ thị chứa cấu trúc con.- 36 sĩ quan Euler: Không tồn tại bảng Latin bậc 6.- Xếp domino: Phủ bằng bảng domino.

5. Lý thuyết tập hợp (Set Theory)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Tập hợp, phần tử: Phép toán hợp, giao, hiệu, bổ sung.- Quan hệ, hàm: Quan hệ là tập con tích Descartes; hàm ánh xạ duy nhất.- Số đếm, số thứ tự: Độ lớn, cấu trúc thứ tự.- Tiên đề ZFC: Tránh nghịch lý Russell.- Lý thuyết mô hình, giả thuyết Continuum: Nghiên cứu cấu trúc toán học.
Bài toán trọng tâm	<ul style="list-style-type: none">- Phân loại tập hợp dựa trên số đếm, thứ tự.- Chứng minh tính chất tập hợp, hàm, quan hệ.- Giải bài toán tính đếm được, ánh xạ tập hợp.- Nghiên cứu cấu trúc toán học qua lý thuyết mô hình.
Câu hỏi lớn	<ul style="list-style-type: none">- Xây dựng nền tảng toán học không mâu thuẫn thế nào?- Có tập hợp nào giữa số tự nhiên và số thực không? (Giả thuyết Continuum)- Mô tả khái niệm trừu tượng trong tính toán ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Ngữ nghĩa ngôn ngữ lập trình: Định nghĩa kiểu dữ liệu.- Logic, kiểm chứng hình thức: Nền tảng logic vị từ.- Lý thuyết tính toán: Định nghĩa ngôn ngữ, máy Turing.
Bài toán kinh điển	<ul style="list-style-type: none">- Đếm tập hợp: So sánh số đếm tập hợp vô hạn.- Giả thuyết Continuum: Có tập hợp giữa \aleph_0 và 2^{\aleph_0} không?- Ánh xạ song ánh: Chứng minh hai tập có cùng số đếm.- Cantor: So sánh lực lượng tập hợp.- Axiom of Choice: Chọn phần tử từ hệ tập.

6. Xác suất rời rạc (Discrete Probability)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Biến ngẫu nhiên: Ánh xạ từ không gian mẫu sang số thực.- Kỳ vọng, phương sai: Giá trị trung bình, độ phân tán.- Phân phối rời rạc: Bernoulli, Binomial, Poisson, Geometric.- Bất đẳng thức: Markov, Chebyshev, Chernoff.- Chuỗi Markov: Quá trình ngẫu nhiên không nhớ.
Bài toán trọng tâm	<ul style="list-style-type: none">- Tính xác suất sự kiện trong không gian mẫu rời rạc.- Phân tích thuật toán ngẫu nhiên.- Mô hình hóa hệ thống tính toán ngẫu nhiên (giao thức mạng, Monte Carlo).- Dự đoán hành vi chuỗi Markov.
Câu hỏi lớn	<ul style="list-style-type: none">- Dự đoán, kiểm soát hệ thống ngẫu nhiên thế nào?- Thuật toán ngẫu nhiên hiệu quả hơn xác định khi nào?- Phân tích độ tin cậy, hiệu suất hệ thống tính toán ra sao?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Thuật toán ngẫu nhiên: QuickSort, kiểm tra nguyên tố.- Học máy lý thuyết: Mô hình học thống kê.- Phân tích mạng: Mô hình lưu lượng, độ trễ bằng chuỗi Markov.
Bài toán kinh điển	<ul style="list-style-type: none">- Phân tích thuật toán ngẫu nhiên: Phân tích QuickSort, Miller-Rabin.- Chuỗi Markov: Tính trạng thái ổn định.- Chernoff: Phân tích xác suất sự kiện ngẫu nhiên.- Sinh nhật: Xác suất trùng ngày sinh.- Monty Hall: Đổi cửa tăng cơ hội thắng.

7. Lý thuyết đồ thị (Graph Theory)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Đồ thị: Đỉnh, cạnh (hướng/vô hướng).- Cây: Đồ thị không chu trình.- Đường đi, chu trình: Đường đi ngắn nhất, chu trình Euler, Hamilton.- Luồng, ghép đôi: Tối ưu luồng, ghép đôi tối đa.- Lý thuyết Ramsey: Tính chất tổ hợp đồ thị lớn.
Bài toán trọng tâm	<ul style="list-style-type: none">- Tìm đường đi ngắn nhất, luồng tối đa.- Xác định liên thông, chu trình, số màu.

Hạng mục	Nội dung
	<ul style="list-style-type: none">- Giải bài toán ghép đôi, bao phủ, phân hoạch.- Phân tích tính chất tổ hợp đồ thị.
Câu hỏi lớn	<ul style="list-style-type: none">- Mô hình hóa, tối ưu hóa hệ thống phức tạp (mạng, giao thông) bằng đồ thị thế nào?- Cải thiện độ phức tạp thuật toán đồ thị ra sao?- Cấu trúc tổ hợp đồ thị tiết lộ gì về hệ thống tính toán?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Mạng máy tính: Tối ưu định tuyến, băng thông.- Thuật toán: Dijkstra, Kruskal.- Tối ưu tổ hợp: Lập lịch, phân bổ tài nguyên.
Bài toán kinh điển	<ul style="list-style-type: none">- Ghép đôi tối đa: Tìm tập cạnh lớn nhất không chung đỉnh.- Tô màu đồ thị: Tô đỉnh với số màu ít nhất.- Đồng cấu đồ thị: Kiểm tra hai đồ thị có cùng cấu trúc.- Cầu Königsberg: Không tồn tại đường đi Euler.- Người đưa thư Trung Quốc: Chu trình qua mọi cạnh với chi phí thấp nhất.- 4 màu: Tô bản đồ với tối đa 4 màu.

8. Lý thuyết số (Number Theory)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Số nguyên tố, chia hết: Đồng dư, GCD, thuật toán Euclid.- Hàm số học: Euler, Mobius.- Đồng dư: Giải phương trình đồng dư, định lý Fermat, Euler.- Mật mã số học: Trường hữu hạn, nhóm cyclic.- Phân tích số học: Dãy số (Fibonacci, nguyên tố).
Bài toán trọng tâm	<ul style="list-style-type: none">- Kiểm tra nguyên tố, phân tích thừa số nguyên tố.- Giải phương trình đồng dư.- Thiết kế giao thức mật mã (RSA, Diffie-Hellman).- Phân tích dãy số trong tổ hợp.
Câu hỏi lớn	<ul style="list-style-type: none">- Bảo mật thông tin bằng tính chất số học thế nào?- Số nguyên tố phân bố ra sao? (Riemann Hypothesis)- Cấu trúc số học giải bài toán tính toán thế nào?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Mật mã học: RSA, mã hóa đường cong elliptic.- Thuật toán: Phân tích số, GCD.- Lý thuyết mã hóa: Mã sửa lỗi trên trường hữu hạn.
Bài toán kinh điển	<ul style="list-style-type: none">- Kiểm tra nguyên tố: Xác định số nguyên tố (Miller-Rabin, AKS).- Phân tích số: Phân tích số thành thừa số nguyên tố.- Đồng dư tuyến tính: Giải $ax \equiv b \pmod m$.- Fermat cuối cùng: Không nghiệm nguyên cho $x^n + y^n = z^n$, $n > 2$.- Số hoàn hảo: Tổng ước bằng chính nó.

9. Hình học rời rạc (Discrete Geometry)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Điểm lưới: Tọa độ nguyên.- Bao lồi: Tập lồi nhỏ nhất chứa các điểm.- Đa giác, phân hoạch: Phân chia không gian rời rạc.- Định lý Pick: Liên hệ diện tích, điểm lưới.- Thuật toán hình học: Giao điểm, khoảng cách, bao lồi.
Bài toán trọng tâm	<ul style="list-style-type: none">- Tính diện tích, chu vi, số điểm lưới.- Tìm bao lồi, phân hoạch tối ưu.- Giải bài toán tối ưu hình học (Traveling Salesman trên lưới).- Mô hình hóa cấu trúc không gian (vi mạch, đồ họa).
Câu hỏi lớn	<ul style="list-style-type: none">- Mô tả, tối ưu hóa cấu trúc không gian rời rạc thế nào?- Thuật toán hình học đạt độ phức tạp nào?- Cấu trúc hình học tiết lộ gì về thiết kế hệ thống tính toán?
💡Ứng dụng trong TCS	<ul style="list-style-type: none">- Thiết kế vi mạch: Sắp xếp thành phần.- Đồ họa máy tính: Tính bao lồi, phân hoạch.- Hệ thống phân tán: Mô hình vị trí, kết nối nút mạng.

Hạng mục	Nội dung
Bài toán kinh điển	<ul style="list-style-type: none">- Bao lồi: Tìm đa giác lồi nhỏ nhất chứa các điểm.- Người bán hàng (TSP): Tìm hành trình ngắn nhất qua các điểm.- Định lý Pick: Tính diện tích đa giác trên lưới.- Đóng gói hình tròn: Sắp xếp hình tròn tối ưu.- Erdős–Szekeres: Tìm ngũ giác lồi trong tập điểm.

10. Lý thuyết tính toán (Computability Theory)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Máy Turing: Mô hình tính toán lý thuyết.- Ngôn ngữ hình thức: Chuỗi ký tự, ngữ pháp, automata.- Tính quyết định: Bài toán có thuật toán giải không.- Bài toán dừng: Không quyết định được.- Độ phức tạp: Phân loại bài toán theo tài nguyên.
Bài toán trọng tâm	<ul style="list-style-type: none">- Xác định ngôn ngữ/bài toán có quyết định bởi máy Turing không.- Phân loại bài toán theo tính tính toán được.- Nghiên cứu giới hạn tính toán (bài toán dừng).- Phân tích độ phức tạp bài toán.
Câu hỏi lớn	<ul style="list-style-type: none">- Bài toán nào máy tính giải được/không giải được?- Phân loại bài toán theo độ khó tính toán thế nào?- Giới hạn tính toán tiết lộ gì về trí tuệ, máy móc?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Kiểm chứng hình thức: Kiểm tra tính đúng đắn hệ thống.- Thiết kế ngôn ngữ lập trình: Ngôn ngữ hình thức cho trình biên dịch.- AI lý thuyết: Khả năng tính toán của mô hình học máy.
Bài toán kinh điển	<ul style="list-style-type: none">- Dừng (Halting): Kiểm tra chương trình có dừng không.- Ngôn ngữ hình thức: Kiểm tra ngôn ngữ được chấp nhận bởi automata.- P vs NP: Liệu $P = NP$?- Post Correspondence: Không quyết định được.- Máy Turing: Mô hình hóa tính toán.

11. Lý thuyết phạm trù (Category Theory)

Hạng mục	Nội dung
Khái niệm cốt lõi	<ul style="list-style-type: none">- Phạm trù: Đối tượng, mũi tên thỏa mãn tính kết hợp, đơn vị.- Hàm tử: Ánh xạ bảo toàn cấu trúc.- Biến đổi tự nhiên: Ánh xạ giữa hàm tử.- Tính đối ngẫu: Quan hệ khái niệm đối lập.- Cấu trúc đặc biệt: Sản phẩm, coproduct, pullback, pushout.
Bài toán trọng tâm	<ul style="list-style-type: none">- Mô tả cấu trúc toán học, tính toán bằng phạm trù.- Tìm hàm tử, biến đổi tự nhiên liên kết cấu trúc.- Nghiên cứu đối ngẫu để khám phá quan hệ.- Tổng quát hóa khái niệm trong logic, đại số, tính toán.
Câu hỏi lớn	<ul style="list-style-type: none">- Thống nhất toán học, tính toán bằng ngôn ngữ chung thế nào?- Mô tả, phân tích cấu trúc tính toán qua phạm trù ra sao?- Phạm trù đơn giản hóa, khám phá khái niệm mới trong TCS thế nào?
Ứng dụng trong TCS	<ul style="list-style-type: none">- Ngữ nghĩa ngôn ngữ lập trình: Mô tả kiểu dữ liệu, chương trình.- Lý thuyết loại: Nền tảng hệ thống loại trong lập trình hàm.- Formal methods: Kiểm chứng, thiết kế hệ thống.
Bài toán kinh điển	<ul style="list-style-type: none">- Tìm hàm tử: Xây dựng hàm tử bảo toàn cấu trúc.- Biến đổi tự nhiên: Tìm biến đổi giữa hai hàm tử.- Đối ngẫu phạm trù: Khám phá quan hệ sản phẩm, coproduct.- Yoneda Lemma: Đối tượng xác định qua ánh xạ.- Giới hạn/đối giới hạn: Tìm cấu trúc phổ quát.

a. Đại số trừu tượng (Abstract Algebra)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Nhóm (Groups)	Phân loại cấu trúc đại số (nhóm, vành, trường) dựa trên tính chất.	Làm thế nào để mô tả và phân loại các cấu trúc đại số một cách tổng quát?	Mật mã học (nhóm và trường hữu hạn).
Vành (Rings)	Tìm đồng cấu và đẳng cấu giữa các cấu trúc.	Các cấu trúc đại số có thể được sử dụng như thế nào để giải quyết các bài toán trong tính toán (mã hóa, tối ưu hóa)?	Lý thuyết mã hóa (vành và trường để thiết kế mã sửa lỗi).
Trường (Fields)	Giải các bài toán liên quan đến đối xứng (nhóm hoán vị, nhóm đối xứng của đa giác).	Làm thế nào để ánh xạ giữa các cấu trúc đại số giúp hiểu các hệ thống phức tạp hơn (lý thuyết tính toán, vật lý lý thuyết)?	Thiết kế thuật toán (cấu trúc nhóm trong thuật toán đồ thị và tối ưu hóa).
Đồng cấu (Homomorphisms)	Ứng dụng trong mật mã học (mã hóa RSA, dựa trên vành và trường).		
Lý thuyết Galois (Galois Theory)			

b. Logic

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Logic mệnh đề (Propositional Logic)	Xây dựng các hệ thống logic để mô tả và kiểm chứng các hệ thống tính toán.	Làm thế nào để biểu diễn tri thức và lập luận một cách chính xác trong các hệ thống máy tính?	Kiểm chứng hình thức (sử dụng logic để kiểm tra tính đúng đắn của phần cứng/phần mềm).
Logic vị từ (Predicate Logic)	Chứng minh hoặc bác bỏ các công thức logic trong một hệ thống nhất định.	Liệu một công thức logic có thể được chứng minh hoặc bác bỏ trong một hệ thống nhất định không? (Vấn đề quyết định)	Trí tuệ nhân tạo (logic là nền tảng cho các hệ thống suy luận và biểu diễn tri thức).
Lý thuyết mô hình (Model Theory)	Kiểm tra tính thỏa mãn (satisfiability) của các công thức logic (SAT problem).	Làm thế nào để tự động hóa các quá trình suy luận logic trong các ứng dụng thực tế (AI, kiểm chứng phần mềm)?	Ngữ nghĩa ngôn ngữ lập trình (logic giúp định nghĩa ý nghĩa của các chương trình).
Lý thuyết chứng minh (Proof Theory)	Thiết kế các công cụ tự động hóa suy luận logic (trình giải SAT, trình kiểm chứng mô hình).		
Tính toán logic (Computational Logic)			

c. Toán rời rạc (Discrete Mathematics)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Quan hệ và hàm (Relations and Functions)	Phân tích và thiết kế thuật toán trên các cấu trúc rời rạc (đồ thị, cây).	Làm thế nào để mô hình hóa và giải quyết các bài toán thực tế bằng các cấu trúc rời rạc?	Thiết kế thuật toán (toán rời rạc cung cấp công cụ để phân tích độ phức tạp và hiệu suất).
Đồ thị và cây (Graphs and Trees)	Đếm số lượng cấu hình hoặc tổ hợp trong một hệ thống.	Các thuật toán trên cấu trúc rời rạc có thể được tối ưu hóa đến mức nào?	Mạng máy tính (đồ thị mô hình hóa các kết nối mạng).
Tổ hợp đếm (Combinatorics)	Tối ưu hóa trên các cấu trúc rời rạc (bài toán đường đi ngắn nhất, luồng tối đa).	Làm thế nào để đếm hoặc liệt kê tất cả các cấu hình khả thi trong một hệ thống phức tạp?	Tối ưu hóa (các bài toán tổ hợp xuất hiện trong lập lịch, phân bổ tài nguyên).
Lý thuyết tập hợp rời rạc (Discrete Set Theory)	Mô hình hóa các hệ thống tính toán bằng các cấu trúc rời rạc.		
Thuật toán cơ bản (Basic Algorithms)			

d. Tổ hợp (Combinatorics)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
-------------------	--------------------	-------------	--------------------

Đếm (Counting)	Đếm số cách sắp xếp hoặc chọn các phần tử thỏa mãn các ràng buộc.	Có bao nhiêu cách để thực hiện một nhiệm vụ hoặc cấu hình một hệ thống?	Phân tích thuật toán (tổ hợp giúp tính toán số phép toán cần thiết).
Nguyên lý bao hàm-loại trừ (Inclusion-Exclusion)	Phân tích các cấu trúc tổ hợp trong đồ thị (số vòng, số cây).	Làm thế nào để tối ưu hóa các cấu trúc tổ hợp trong các bài toán thực tế?	Mật mã học (tổ hợp được sử dụng trong thiết kế mã và giao thức).
Hàm sinh (Generating Functions)	Thiết kế các cấu trúc tổ hợp cho các ứng dụng thực tế (mã hóa, lập lịch).	Các cấu trúc tổ hợp có thể được tổng quát hóa hoặc phân loại như thế nào?	Tối ưu hóa (các bài toán tổ hợp xuất hiện trong lập kế hoạch và phân bổ tài nguyên).
Tổ hợp đồ thị (Combinatorial Graph Theory)	Giải các bài toán tối ưu tổ hợp (bài toán ghép đôi, bao phủ).		
Thiết kế tổ hợp (Combinatorial Design)			

e. Lý thuyết tập hợp (Set Theory)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Tập hợp và phần tử (Sets and Elements)	Xác định và phân loại các tập hợp dựa trên kích thước (số đếm) và cấu trúc (thứ tự).	Làm thế nào để xây dựng một nền tảng toán học không mâu thuẫn để mô tả mọi cấu trúc toán học?	Ngữ nghĩa ngôn ngữ lập trình (lý thuyết tập hợp cung cấp ngôn ngữ để định nghĩa các kiểu dữ liệu và cấu trúc).
Quan hệ và hàm (Relations and Functions)	Chứng minh các tính chất của tập hợp, hàm, và quan hệ.	Các tập hợp có kích thước nào tồn tại giữa số đếm của số tự nhiên và số thực? (Giả thuyết Continuum)	Logic và kiểm chứng hình thức (lý thuyết tập hợp là nền tảng cho logic vị từ và lý thuyết mô hình).
Số đếm và số thứ tự (Cardinals and Ordinals)	Giải quyết các bài toán liên quan đến tính đếm được và các ánh xạ giữa các tập hợp.	Làm thế nào để sử dụng tập hợp để mô tả các khái niệm trừu tượng trong tính toán (ngôn ngữ lập trình, máy Turing)?	Lý thuyết tính toán (dùng để định nghĩa các tập hợp ngôn ngữ và máy Turing).
Lý thuyết tập hợp tiên đề (Axiomatic Set Theory)	Nghiên cứu các cấu trúc toán học thông qua lý thuyết mô hình.		
Lý thuyết mô hình và giả thuyết Continuum			

f. Xác suất rời rạc (Discrete Probability)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Biến ngẫu nhiên (Random Variables)	Tính xác suất của các sự kiện trong không gian mẫu rời rạc.	Làm thế nào để dự đoán hoặc kiểm soát hành vi của các hệ thống ngẫu nhiên?	Thuật toán ngẫu nhiên (QuickSort ngẫu nhiên, kiểm tra tính nguyên tố).
Kỳ vọng và phương sai (Expectation and Variance)	Phân tích hiệu suất của các thuật toán ngẫu nhiên (randomized algorithms).	Các thuật toán ngẫu nhiên có thể hiệu quả hơn các thuật toán xác định trong trường hợp nào?	Học máy lý thuyết (xác suất rời rạc là nền tảng cho các mô hình học thống kê).
Phân phối rời rạc (Discrete Distributions)	Mô hình hóa các hệ thống tính toán có yếu tố ngẫu nhiên.	Làm thế nào để phân tích độ tin cậy hoặc hiệu suất của các hệ thống tính toán dựa trên xác suất?	Phân tích mạng (mô hình hóa lưu lượng mạng hoặc độ trễ bằng chuỗi Markov).
Bất đẳng thức xác suất (Probability Inequalities)	Dự đoán hành vi lâu dài của các quá trình ngẫu nhiên (chuỗi Markov).		
Chuỗi Markov (Markov Chains)			

g. Lý thuyết đồ thị (Graph Theory)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Đồ thị (Graphs)	Tìm đường đi ngắn nhất (Shortest Path) hoặc luồng tối đa (Maximum Flow) trong mạng.	Làm thế nào để mô hình hóa và tối ưu hóa các hệ thống phức tạp bằng đồ thị?	Mạng máy tính (mô hình hóa và tối ưu hóa định tuyến, băng thông).

Cây (Trees)	Xác định tính liên thông, tính chất chu trình, hoặc tính màu (chromatic number) của đồ thị.	Các thuật toán trên đồ thị có thể được cải thiện về độ phức tạp như thế nào?	Thuật toán (các thuật toán đồ thị như Dijkstra, Kruskal là cốt lõi của nhiều bài toán TCS).
Đường đi và chu trình (Paths and Cycles)	Giải các bài toán ghép đôi, bao phủ, hoặc phân hoạch đồ thị.	Các cấu trúc tổ hợp trong đồ thị tiết lộ điều gì về tính chất của các hệ thống tính toán?	Tối ưu hóa tổ hợp (ứng dụng trong lập lịch, phân bổ tài nguyên).
Luồng và ghép đôi (Flows and Matchings)	Phân tích các tính chất tổ hợp của đồ thị.		
Lý thuyết Ramsey (Ramsey Theory)			

h. Lý thuyết số (Number Theory)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Số nguyên tố và tính chia hết (Primes and Divisibility)	Kiểm tra tính nguyên tố và phân tích số thành thừa số nguyên tố.	Làm thế nào để bảo mật thông tin bằng các tính chất số học?	Mật mã học (RSA, mã hóa đường cong elliptic).
Hàm số học (Arithmetic Functions)	Giải các phương trình đồng dư và hệ phương trình đồng dư.	Các số nguyên tố được phân bố như thế nào trong tập số tự nhiên? (Vấn đề Riemann Hypothesis)	Thuật toán (thuật toán phân tích số, GCD).
Lý thuyết đồng dư (Congruences)	Thiết kế các giao thức mật mã dựa trên các tính chất số học.	Các cấu trúc số học có thể được sử dụng như thế nào để giải quyết các bài toán tính toán?	Lý thuyết mã hóa (mã sửa lỗi dựa trên các trường hữu hạn).
Mật mã số học (Number Theoretic Cryptography)	Phân tích các dãy số hoặc cấu trúc số học trong các bài toán tổ hợp.		
Phân tích số học (Analytic Number Theory)			

i. Hình học rời rạc (Discrete Geometry)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Điểm lưới (Lattice Points)	Tính diện tích, chu vi, hoặc số điểm lưới trong các cấu trúc hình học rời rạc.	Làm thế nào để mô tả và tối ưu hóa các cấu trúc không gian rời rạc?	Thiết kế vi mạch (sắp xếp các thành phần trên bo mạch).
Bao lồi (Convex Hull)	Tìm bao lồi hoặc phân hoạch tối ưu cho tập hợp điểm.	Các thuật toán hình học rời rạc có thể đạt được độ phức tạp nào?	Đồ họa máy tính (tính toán bao lồi và phân hoạch không gian).
Đa giác và phân hoạch (Polygons and Partitions)	Giải các bài toán tối ưu hình học trên lưới.	Các cấu trúc hình học rời rạc tiết lộ điều gì về thiết kế hệ thống tính toán?	Hệ thống phân tán (mô hình hóa vị trí và kết nối của các nút trong mạng).
Định lý Pick (Pick's Theorem)	Mô hình hóa các cấu trúc không gian trong thiết kế vi mạch hoặc đồ họa máy tính.		
Thuật toán hình học (Geometric Algorithms)			

k. Lý thuyết tính toán (Computability Theory)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Máy Turing (Turing Machines)	Xác định liệu một ngôn ngữ hoặc bài toán có thể được quyết định bởi máy Turing.	Những bài toán nào có thể được giải quyết bằng máy tính, và những bài toán nào không thể?	Kiểm chứng hình thức (dùng lý thuyết tính toán để kiểm tra tính đúng đắn của hệ thống).
Ngôn ngữ hình thức (Formal Languages)	Phân loại các bài toán theo tính tính toán được hoặc không tính toán được.	Làm thế nào để phân loại các bài toán theo mức độ khó về tính toán?	Thiết kế ngôn ngữ lập trình (ngôn ngữ hình thức là nền tảng cho trình biên dịch).

Tính quyết định (Decidability)	Nghiên cứu các giới hạn của tính toán (bài toán dừng, bài toán logic).	Các giới hạn của tính toán tiết lộ điều gì về bản chất của trí tuệ và máy móc?	AI lý thuyết (nghiên cứu khả năng tính toán của các mô hình học máy).
Bài toán dừng (Halting Problem)	Phân tích độ phức tạp của các bài toán quyết định được.		
Độ phức tạp (Complexity)			

I. Lý thuyết phạm trù (Category Theory)

Khái niệm cốt lõi	Bài toán trọng tâm	Câu hỏi lớn	Ứng dụng trong TCS
Phạm trù (Categories)	Mô tả các cấu trúc toán học và tính toán bằng ngôn ngữ phạm trù.	Làm thế nào để thống nhất các lĩnh vực toán học và tính toán bằng một ngôn ngữ chung?	Ngữ nghĩa ngôn ngữ lập trình (dùng phạm trù để mô tả các kiểu dữ liệu và chương trình).
Hàm tử (Functors)	Tìm các hàm tử và biến đổi tự nhiên để liên kết các cấu trúc khác nhau.	Các cấu trúc tính toán có thể được mô tả và phân tích như thế nào qua phạm trù?	Lý thuyết loại (lý thuyết

a. Đại số trừu tượng (Abstract Algebra)

- Khái niệm cốt lõi:**
 - Nhóm (Groups):** Cấu trúc với một phép toán kết hợp, có phần tử đơn vị và nghịch đảo (ví dụ: phép cộng trên số nguyên, phép nhân trên số thực khác 0).
 - Vành (Rings):** Cấu trúc với hai phép toán (cộng và nhân), ví dụ: số nguyên với cộng và nhân thông thường.
 - Trường (Fields):** Vành mà mọi phần tử khác 0 đều có nghịch đảo nhân, ví dụ: số thực, số phức.
 - Đồng cấu (Homomorphisms):** Hàm bảo toàn cấu trúc giữa các đại số.
 - Lý thuyết Galois:** Liên kết giữa các trường mở rộng và nhóm đối xứng.
- Bài toán trọng tâm:**
 - Phân loại các cấu trúc đại số (nhóm, vành, trường) dựa trên tính chất của chúng.
 - Tìm các đồng cấu và đẳng cấu giữa các cấu trúc.
 - Giải các bài toán liên quan đến đối xứng (ví dụ: nhóm hoán vị, nhóm đối xứng của đa giác).
 - Ứng dụng trong mật mã học (như mã hóa RSA, dựa trên vành và trường).
- Câu hỏi lớn:**
 - Làm thế nào để mô tả và phân loại các cấu trúc đại số một cách tổng quát?
 - Các cấu trúc đại số có thể được sử dụng như thế nào để giải quyết các bài toán trong tính toán (ví dụ: mã hóa, tối ưu hóa)?
 - Làm thế nào để ánh xạ giữa các cấu trúc đại số giúp hiểu các hệ thống phức tạp hơn (như trong lý thuyết tính toán hoặc vật lý lý thuyết)?
- Ứng dụng trong TCS:**
 - Mật mã học: Sử dụng các nhóm và trường hữu hạn.
 - Lý thuyết mã hóa (Coding Theory): Dựa trên vành và trường để thiết kế mã sửa lỗi.
 - Thiết kế thuật toán: Các cấu trúc nhóm xuất hiện trong các thuật toán đồ thị và tối ưu hóa.

b. Logic

- Khái niệm cốt lõi:**
 - Logic mệnh đề (Propositional Logic):** Xử lý các mệnh đề (đúng/sai) và các phép nối logic (AND, OR, NOT).
 - Logic vị từ (Predicate Logic):** Mở rộng logic mệnh đề với các biến và lượng từ (\forall , \exists).
 - Lý thuyết mô hình (Model Theory):** Nghiên cứu mối quan hệ giữa các công thức logic và cấu trúc toán học.
 - Lý thuyết chứng minh (Proof Theory):** Phân tích các phương pháp chứng minh và tính đúng đắn của chúng.
 - Tính toán logic (Computational Logic):** Logic trong các hệ thống tính toán, như logic tạm thời (temporal logic) hoặc logic mô tả (description logic).
- Bài toán trọng tâm:**
 - Xây dựng các hệ thống logic để mô tả và kiểm chứng các hệ thống tính toán.
 - Chứng minh hoặc bác bỏ các công thức logic trong một hệ thống nhất định.
 - Kiểm tra tính thỏa mãn (satisfiability) của các công thức logic (SAT problem).
 - Thiết kế các công cụ tự động hóa suy luận logic (như trình giải SAT, trình kiểm chứng mô hình).
- Câu hỏi lớn:**

- Làm thế nào để biểu diễn tri thức và lập luận một cách chính xác trong các hệ thống máy tính?
- Liệu một công thức logic có thể được chứng minh hoặc bác bỏ trong một hệ thống nhất định không? (Vấn đề quyết định)
- Làm thế nào để tự động hóa các quá trình suy luận logic trong các ứng dụng thực tế (như AI, kiểm chứng phần mềm)?
- Ứng dụng trong TCS:**
 - Kiểm chứng hình thức (Formal Verification): Sử dụng logic để kiểm tra tính đúng đắn của phần cứng/phần mềm.
 - Trí tuệ nhân tạo: Logic là nền tảng cho các hệ thống suy luận và biểu diễn tri thức.
 - Ngữ nghĩa ngôn ngữ lập trình: Logic giúp định nghĩa ý nghĩa của các chương trình.

c. Toán rời rạc (Discrete Mathematics)

- Khái niệm cốt lõi:**
 - Quan hệ và hàm (Relations and Functions):** Mô tả các mối liên hệ giữa các tập hợp.
 - Đồ thị và cây (Graphs and Trees):** Cấu trúc để mô hình hóa mạng, hệ thống, hoặc thuật toán.
 - Tổ hợp đếm (Combinatorics):** Đếm số cách sắp xếp hoặc lựa chọn các đối tượng.
 - Lý thuyết tập hợp rời rạc:** Các tập hợp hữu hạn và các phép toán trên chúng.
 - Thuật toán cơ bản:** Phân tích và thiết kế thuật toán trên các cấu trúc rời rạc.
- Bài toán trọng tâm:**
 - Phân tích và thiết kế thuật toán trên các cấu trúc rời rạc (như đồ thị, cây).
 - Đếm số lượng cấu hình hoặc tổ hợp trong một hệ thống (ví dụ: số cách xếp n vật).
 - Tối ưu hóa trên các cấu trúc rời rạc (như bài toán đường đi ngắn nhất, luồng tối đa).
 - Mô hình hóa các hệ thống tính toán bằng các cấu trúc rời rạc.
- Câu hỏi lớn:**
 - Làm thế nào để mô hình hóa và giải quyết các bài toán thực tế bằng các cấu trúc rời rạc?
 - Các thuật toán trên cấu trúc rời rạc có thể được tối ưu hóa đến mức nào?
 - Làm thế nào để đếm hoặc liệt kê tất cả các cấu hình khả thi trong một hệ thống phức tạp?
- Ứng dụng trong TCS:**
 - Thiết kế thuật toán: Toán rời rạc cung cấp các công cụ để phân tích độ phức tạp và hiệu suất.
 - Mạng máy tính: Đồ thị mô hình hóa các kết nối mạng.
 - Tối ưu hóa: Các bài toán tổ hợp xuất hiện trong lập lịch, phân bổ tài nguyên.

d. Tổ hợp (Combinatorics)

- Khái niệm cốt lõi:**
 - Đếm (Counting):** Tính số cách chọn, sắp xếp, hoặc phân bố các đối tượng (phép đếm, hoán vị, tổ hợp).
 - Nguyên lý bao hàm-loại trừ (Inclusion-Exclusion):** Đếm các tập hợp phức tạp.
 - Hàm sinh (Generating Functions):** Công cụ để giải các bài toán đếm và phân tích chuỗi.
 - Tổ hợp đồ thị (Combinatorial Graph Theory):** Nghiên cứu các tính chất tổ hợp của đồ thị.
 - Thiết kế tổ hợp (Combinatorial Design):** Xây dựng các cấu trúc như khối hoặc ma trận.
- Bài toán trọng tâm:**
 - Đếm số cách sắp xếp hoặc chọn các phần tử thỏa mãn các ràng buộc.
 - Phân tích các cấu trúc tổ hợp trong đồ thị (như số vòng, số cây).
 - Thiết kế các cấu trúc tổ hợp cho các ứng dụng thực tế (như mã hóa, lập lịch).
 - Giải các bài toán tối ưu tổ hợp (như bài toán ghép đôi, bao phủ).
- Câu hỏi lớn:**
 - Có bao nhiêu cách để thực hiện một nhiệm vụ hoặc cấu hình một hệ thống?
 - Làm thế nào để tối ưu hóa các cấu trúc tổ hợp trong các bài toán thực tế?
 - Các cấu trúc tổ hợp có thể được tổng quát hóa hoặc phân loại như thế nào?
- Ứng dụng trong TCS:**
 - Phân tích thuật toán: Tổ hợp giúp tính toán số phép toán cần thiết.
 - Mật mã học: Tổ hợp được sử dụng trong thiết kế mã và giao thức.
 - Tối ưu hóa: Các bài toán tổ hợp xuất hiện trong lập kế hoạch và phân bổ tài nguyên.

e. Lý thuyết tập hợp (Set Theory)

- Khái niệm cốt lõi:**
 - Tập hợp và phần tử:** Tập hợp là tập hợp các đối tượng (phần tử) được xác định rõ ràng; các phép toán cơ bản như hợp, giao, hiệu, bổ sung.
 - Quan hệ và hàm:** Quan hệ là tập con của tích Descartes; hàm là quan hệ đặc biệt ánh xạ mỗi phần tử đến duy nhất một phần tử khác.

- Số đếm và số thứ tự (Cardinals and Ordinals):** Độ lớn của tập hợp (đếm được, không đếm được) và cấu trúc thứ tự.
- Lý thuyết tập hợp tiên đề (Axiomatic Set Theory):** Hệ tiên đề Zermelo-Fraenkel (ZF) hoặc ZF với tiên đề lựa chọn (ZFC) để tránh các nghịch lý (như nghịch lý Russell).
- Lý thuyết mô hình và giả thuyết Continuum:** Nghiên cứu các cấu trúc toán học thỏa mãn các công thức logic.
- Bài toán trọng tâm:**
 - Xác định và phân loại các tập hợp dựa trên kích thước (số đếm) và cấu trúc (thứ tự).
 - Chứng minh các tính chất của tập hợp, hàm, và quan hệ (ví dụ: tính tương đương, tính phản xạ).
 - Giải quyết các bài toán liên quan đến tính đếm được (countability) và các ánh xạ giữa các tập hợp.
 - Nghiên cứu các cấu trúc toán học thông qua lý thuyết mô hình, chẳng hạn như mô hình của số học hoặc đại số.
- Câu hỏi lớn:**
 - Làm thế nào để xây dựng một nền tảng toán học không mâu thuẫn để mô tả mọi cấu trúc toán học?
 - Các tập hợp có kích thước nào tồn tại giữa số đếm của số tự nhiên và số thực? (Giả thuyết Continuum)
 - Làm thế nào để sử dụng tập hợp để mô tả các khái niệm trừu tượng trong tính toán, như ngôn ngữ lập trình hoặc máy Turing?
- Ứng dụng trong TCS:**
 - Ngữ nghĩa ngôn ngữ lập trình:** Lý thuyết tập hợp cung cấp ngôn ngữ để định nghĩa các kiểu dữ liệu và cấu trúc.
 - Logic và kiểm chứng hình thức:** Lý thuyết tập hợp là nền tảng cho logic vị từ và lý thuyết mô hình.
 - Lý thuyết tính toán:** Dùng để định nghĩa các tập hợp ngôn ngữ và máy Turing.

f. Xác suất rời rạc (Discrete Probability)

- Khái niệm cốt lõi:**
 - Biến ngẫu nhiên (Random Variables):** Ánh xạ từ không gian mẫu sang số thực, với phân phối xác suất.
 - Kỳ vọng và phương sai (Expectation and Variance):** Đo lường giá trị trung bình và độ phân tán của biến ngẫu nhiên.
 - Phân phối rời rạc:** Bernoulli, Binomial, Poisson, Geometric.
 - Bất đẳng thức xác suất:** Markov, Chebyshev, Chernoff bounds để phân tích hành vi ngẫu nhiên.
 - Chuỗi Markov:** Mô hình các quá trình ngẫu nhiên có tính chất "không nhớ" (memoryless).
- Bài toán trọng tâm:**
 - Tính xác suất của các sự kiện trong không gian mẫu rời rạc.
 - Phân tích hiệu suất của các thuật toán ngẫu nhiên (randomized algorithms).
 - Mô hình hóa các hệ thống tính toán có yếu tố ngẫu nhiên, như giao thức mạng hoặc thuật toán Monte Carlo.
 - Dự đoán hành vi lâu dài của các quá trình ngẫu nhiên (như chuỗi Markov).
- Câu hỏi lớn:**
 - Làm thế nào để dự đoán hoặc kiểm soát hành vi của các hệ thống ngẫu nhiên?
 - Các thuật toán ngẫu nhiên có thể hiệu quả hơn các thuật toán xác định trong trường hợp nào?
 - Làm thế nào để phân tích độ tin cậy hoặc hiệu suất của các hệ thống tính toán dựa trên xác suất?
- Ứng dụng trong TCS:**
 - Thuật toán ngẫu nhiên:** Như thuật toán QuickSort ngẫu nhiên hoặc kiểm tra tính nguyên tố (Primality Testing).
 - Học máy lý thuyết:** Xác suất rời rạc là nền tảng cho các mô hình học thống kê.
 - Phân tích mạng:** Mô hình hóa lưu lượng mạng hoặc độ trễ bằng chuỗi Markov.

g. Lý thuyết đồ thị (Graph Theory)

- Khái niệm cốt lõi:**
 - Đồ thị (Graphs):** Tập hợp các đỉnh (vertices) và cạnh (edges), có thể có hướng hoặc vô hướng.
 - Cây (Trees):** Đồ thị không có chu trình, thường dùng để mô hình hóa cấu trúc phân cấp.
 - Đường đi và chu trình (Paths and Cycles):** Tìm đường đi ngắn nhất, chu trình Euler, chu trình Hamilton.
 - Luồng và ghép đôi (Flows and Matchings):** Tối ưu hóa luồng trong mạng hoặc ghép đôi tối đa.
 - Lý thuyết Ramsey:** Nghiên cứu các tính chất tổ hợp của đồ thị lớn.
- Bài toán trọng tâm:**
 - Tìm đường đi ngắn nhất (Shortest Path) hoặc luồng tối đa (Maximum Flow) trong mạng.
 - Xác định tính liên thông, tính chất chu trình, hoặc tính màu (chromatic number) của đồ thị.
 - Giải các bài toán ghép đôi, bao phủ, hoặc phân hoạch đồ thị.
 - Phân tích các tính chất tổ hợp của đồ thị, như số cạnh tối đa mà không tạo thành một cấu trúc cụ thể.
- Câu hỏi lớn:**
 - Làm thế nào để mô hình hóa và tối ưu hóa các hệ thống phức tạp (như mạng máy tính, giao thông) bằng đồ thị?
 - Các thuật toán trên đồ thị có thể được cải thiện về độ phức tạp như thế nào?

- Các cấu trúc tổ hợp trong đồ thị tiết lộ điều gì về tính chất của các hệ thống tính toán?
- Ứng dụng trong TCS:**
 - Mạng máy tính:** Mô hình hóa và tối ưu hóa định tuyến, băng thông.
 - Thuật toán:** Các thuật toán đồ thị (như Dijkstra, Kruskal) là cốt lõi của nhiều bài toán TCS.
 - Tối ưu hóa tổ hợp:** Ứng dụng trong lập lịch, phân bổ tài nguyên.

h. Lý thuyết số (Number Theory)

- Khái niệm cốt lõi:**
 - Số nguyên tố và tính chia hết:** Đồng dư, ước chung lớn nhất (GCD), thuật toán Euclid.
 - Hàm số học:** Hàm Euler, hàm Mobius, dùng để phân tích các tính chất số học.
 - Lý thuyết đồng dư (Congruences):** Giải các phương trình đồng dư, định lý Fermat nhỏ, định lý Euler.
 - Mật mã số học:** Sử dụng các cấu trúc như trường hữu hạn, nhóm cyclic.
 - Phân tích số học:** Nghiên cứu các dãy số, như dãy Fibonacci hoặc dãy số nguyên tố.
- Bài toán trọng tâm:**
 - Kiểm tra tính nguyên tố và phân tích số thành thừa số nguyên tố.
 - Giải các phương trình đồng dư và hệ phương trình đồng dư.
 - Thiết kế các giao thức mật mã dựa trên các tính chất số học (như RSA, Diffie-Hellman).
 - Phân tích các dãy số hoặc cấu trúc số học trong các bài toán tổ hợp.
- Câu hỏi lớn:**
 - Làm thế nào để bảo mật thông tin bằng các tính chất số học?
 - Các số nguyên tố được phân bố như thế nào trong tập số tự nhiên? (Vấn đề Riemann Hypothesis)
 - Các cấu trúc số học có thể được sử dụng như thế nào để giải quyết các bài toán tính toán?
- Ứng dụng trong TCS:**
 - Mật mã học:** RSA, mã hóa đường cong elliptic dựa trên lý thuyết số.
 - Thuật toán:** Thuật toán phân tích số (như Sieve of Eratosthenes) hoặc GCD.
 - Lý thuyết mã hóa:** Mã sửa lỗi dựa trên các trường hữu hạn.

i. Hình học rời rạc (Discrete Geometry)

- Khái niệm cốt lõi:**
 - Điểm lưới (Lattice Points):** Các điểm trong không gian có tọa độ nguyên.
 - Bao lồi (Convex Hull):** Tập hợp nhỏ nhất chứa tất cả các điểm và giữ tính lồi.
 - Đa giác và phân hoạch:** Phân chia không gian thành các vùng rời rạc.
 - Định lý Pick:** Liên hệ giữa diện tích và số điểm lưới trong đa giác.
 - Thuật toán hình học:** Tìm giao điểm, khoảng cách gần nhất, hoặc bao lồi.
- Bài toán trọng tâm:**
 - Tính diện tích, chu vi, hoặc số điểm lưới trong các cấu trúc hình học rời rạc.
 - Tìm bao lồi hoặc phân hoạch tối ưu cho tập hợp điểm.
 - Giải các bài toán tối ưu hình học, như bài toán người bán hàng (Traveling Salesman Problem) trên lưới.
 - Mô hình hóa các cấu trúc không gian trong thiết kế vi mạch hoặc đồ họa máy tính.
- Câu hỏi lớn:**
 - Làm thế nào để mô tả và tối ưu hóa các cấu trúc không gian rời rạc?
 - Các thuật toán hình học rời rạc có thể đạt được độ phức tạp nào?
 - Các cấu trúc hình học rời rạc tiết lộ điều gì về thiết kế hệ thống tính toán?
- Ứng dụng trong TCS:**
 - Thiết kế vi mạch:** Sắp xếp các thành phần trên bo mạch bằng hình học rời rạc.
 - Đồ họa máy tính:** Tính toán bao lồi và phân hoạch không gian.
 - Hệ thống phân tán:** Mô hình hóa vị trí và kết nối của các nút trong mạng.

k. Lý thuyết tính toán (Computability Theory)

- Khái niệm cốt lõi:**

- Máy Turing:** Mô hình tính toán lý thuyết, định nghĩa tính tính toán được.
- Ngôn ngữ hình thức (Formal Languages):** Tập hợp các chuỗi ký tự, được định nghĩa bởi ngữ pháp hoặc automata.
- Tính quyết định (Decidability):** Xác định liệu một bài toán có thể được giải bằng thuật toán hay không.
- Bài toán dừng (Halting Problem):** Ví dụ nổi tiếng về bài toán không thể quyết định.
- Độ phức tạp (Complexity):** Phân loại bài toán theo tài nguyên tính toán (thời gian, không gian).
- Bài toán trọng tâm:**
 - Xác định liệu một ngôn ngữ hoặc bài toán có thể được quyết định bởi máy Turing.
 - Phân loại các bài toán theo tính tính toán được hoặc không tính toán được.
 - Nghiên cứu các giới hạn của tính toán, như bài toán dừng hoặc bài toán logic.
 - Phân tích độ phức tạp của các bài toán quyết định được.
- Câu hỏi lớn:**
 - Những bài toán nào có thể được giải quyết bằng máy tính, và những bài toán nào không thể?
 - Làm thế nào để phân loại các bài toán theo mức độ khó về tính toán?
 - Các giới hạn của tính toán tiết lộ điều gì về bản chất của trí tuệ và máy móc?
- Ứng dụng trong TCS:**
 - Kiểm chứng hình thức:** Dùng lý thuyết tính toán để kiểm tra tính đúng đắn của hệ thống.
 - Thiết kế ngôn ngữ lập trình:** Ngôn ngữ hình thức là nền tảng cho trình biên dịch.
 - AI lý thuyết:** Nghiên cứu khả năng tính toán của các mô hình học máy.

I. Lý thuyết phạm trù (Category Theory)

- Khái niệm cốt lõi:**
 - Phạm trù (Categories):** Tập hợp các đối tượng và mũi tên (morphisms) thỏa mãn các tính chất kết hợp và đơn vị.
 - Hàm tử (Functors):** Ánh xạ giữa các phạm trù, bảo toàn cấu trúc.
 - Biến đổi tự nhiên (Natural Transformations):** Ánh xạ giữa các hàm tử.
 - Tính đối ngẫu (Duality):** Mối liên hệ giữa các khái niệm đối lập trong phạm trù.
 - Các cấu trúc đặc biệt:** Sản phẩm, coproduct, pullback, pushout.
- Bài toán trọng tâm:**
 - Mô tả các cấu trúc toán học và tính toán bằng ngôn ngữ phạm trù.
 - Tìm các hàm tử và biến đổi tự nhiên để liên kết các cấu trúc khác nhau.
 - Nghiên cứu tính đối ngẫu để khám phá các mối quan hệ mới giữa các khái niệm.
 - Áp dụng lý thuyết phạm trù để tổng quát hóa các khái niệm trong logic, đại số, hoặc tính toán.
- Câu hỏi lớn:**
 - Làm thế nào để thống nhất các lĩnh vực toán học và tính toán bằng một ngôn ngữ chung?
 - Các cấu trúc tính toán có thể được mô tả và phân tích như thế nào qua phạm trù?
 - Lý thuyết phạm trù có thể giúp đơn giản hóa hoặc khám phá các khái niệm mới trong TCS như thế nào?
- Ứng dụng trong TCS:**
 - Ngữ nghĩa ngôn ngữ lập trình:** Dùng phạm trù để mô tả các kiểu dữ liệu và chương trình.
 - Lý thuyết loại (Type Theory):** Lý thuyết phạm trù là nền tảng cho các hệ thống loại trong lập trình hàm.
 - Formal methods:** Sử dụng phạm trù để kiểm chứng và thiết kế hệ thống.

Môn Học	Tên Bài Toán	Giải Thích
Đại Số Truợu Tượng	Bài toán nhóm con Sylow	Xác định các nhóm con Sylow của một nhóm hữu hạn. Dùng trong phân tích cấu trúc nhóm.
	Phân loại nhóm Abel	Phân loại tất cả nhóm Abel hữu hạn thành tổng trực tiếp của các nhóm cyclic.
	Bài toán Galois	Liên hệ giữa nghiên cứu đa thức và nhóm hoàn vịnh, để quyết định khả năng giải bằng căn.
Logic Toán Học	Bài toán Gödel	Định lý Gödel: Trong một hệ tiếp đủ, luôn có mệnh đề không thể chứng minh hay bác bỏ.
	Bài toán Thompson	Liên quan tới bài toán quyết định trong logic và tính tính quy của hệ thống.
Toán Rời Rạc	Bài toán Hanoi Tower	Di chuyển đĩa giữa các cột tuân theo quy tắc, làm bằng quy hoạch đệ quy.
	Bài toán sắp xếp topo	Sắp xếp các đỉnh trong đồ thị có hướng không chu trình (DAG) theo thứ tự topological.
Tổ Hợp	Bài toán Ramsey	Tìm số lớn nhất sao cho một đồ thị hoặc phủ đều chứa cấu trúc con nhất định.
	Bài toán chia bài	Tính số cách chia bộ bài hoặc chọn đặt theo quy tắc cho trước.

Môn Học	Tên Bài Toán	Giải Thích
Lý Thuyết Tập Hợp	Bài toán Cantor	So sánh lực lượng giữa các tập hợp vô hạn, tập đếm được và không đếm được.
	Bài toán Axiom of Choice	Liên quan đến việc chọn mỗi phần tử từ mỗi tập trong một hệ tập.
Xác Suất Rời Rạc	Bài toán sinh con trước	Tìm sất xuất và quy luật xác suất của chuỗi này xuất hiện trước chuỗi kia.
	Bài toán sinh nhật	Xác suất có hai người trùng ngày sinh trong nhóm người.
Lý Thuyết Đồ Thị	Bài toán Euler	Kiểm tra tồn tại đường đi Euler trong đồ thị (mệnh đề cầu Königsberg).
	Bài toán 4 màu	Mỗi bản đồ đều tối đa 4 màu là đủ để không vùng nào kề nhau màu.
Lý Thuyết Số	Bài toán Fermat cuối cùng	Phương trình $x^n + y^n = z^n$ không có nghiệm nguyên không tầm thường với $n > 2$.
	Bài toán Số Hoàn Hảo	Tìm các số mà tổng các ước bé hơn nó bằng chính nó.
Hình Học Rời Rạc	Bài toán đóng gói	Tính số hộp lý cách để đóng gói hộp/cốc/mô hình vào không gian.
	Bài toán bao phủ	Tìm số hình nhỏ nhất để bao phủ một hình lớn hơn.
Lý Thuyết Tính Toán	Bài toán Turing Machine	Mô hình hoá khả năng tính toán và quyết định.
	Bài toán dừng	Quyết định liệu một CT có dừng không (định lý dừng).
Lý Thuyết Phạm Trù	Bài toán Yoneda Lemma	Xác định cách một đối tượng trong category có thể xác định bằng cách nó liên hệ với đối tượng khác.
	Bài toán Functor đồng nhất	Tìm cách "ánh xạ" hai category sao cho bảo toàn các cấu trúc của chúng.

Môn học	Tên bài toán kinh điển	Mô tả / Giải thích
Đại số trừu tượng	Bài toán đồng cấu (Isomorphism Problem)	Cho hai nhóm, và hỏi liệu có tồn tại một đồng cấu giữa chúng. Rất quan trọng trong việc phân loại các cấu trúc đại số.
	Phân loại nhóm hữu hạn đơn	Việc liệt kê tất cả các nhóm đơn hữu hạn là một kỳ tích toán học thế kỷ 20, dẫn đến “Bảng tuần hoàn của các nhóm đơn”.
	Nhóm Galois	Nghiên cứu tính giải được của phương trình đại số bằng căn thức thông qua nhóm đối xứng các nghiệm.
	Phân tích nhóm thành nhóm con Sylow	Cơ sở để hiểu cấu trúc nội tại của nhóm hữu hạn.
	Logic Toán học	Bài toán quyết định (Entscheidungsproblem)
	Định lý Godel về tính đầy đủ và tính không đầy đủ	Hỏi liệu có thuật toán để xác định mọi công thức logic là đúng hay sai. Được chứng minh là bất khả bởi Turing & Church.
	Bài toán thỏa mãn mệnh đề (SAT)	Kiểm tra xem một công thức logic có thể được gán giá trị đúng/sai như thế nào để trở thành mệnh đề đúng. Là bài toán NP-đầy đủ đầu tiên.
Toán rời rạc	Tô màu bản đồ (Four Color Theorem)	Mọi bản đồ phẳng có thể được tô bằng 4 màu sao cho không vùng kề nhau cùng màu.
	Bài toán bao phủ tập hợp (Set Cover Problem)	Tìm tập con nhỏ nhất của tập con để bao phủ toàn bộ tập gốc.
	Bài toán lịch thi đấu	Sắp xếp các cặp thi đấu sao cho mọi đội gặp nhau 1 lần mà không trùng lịch. Ứng dụng thực tế cao.
Tổ hợp	Bài toán 36 sĩ quan của Euler	Liên quan đến bảng Latin bậc 6, Euler cho rằng không tồn tại – sau này được chứng minh là đúng.
	Bài toán các đường chéo trong đa giác	Số cách nối các đỉnh của đa giác lồi bằng các đường chéo không cắt nhau – liên quan tới số Catalan.
	Bài toán xếp domino	Số cách phủ một bảng bằng các mảnh domino (2×1) – ví dụ bảng 2×n có cách phủ bằng số Fibonacci.
Lý thuyết tập hợp	Nghịch lý Russell	Tập hợp các tập hợp không chứa chính nó – dẫn đến sự ra đời của lý thuyết tập hợp Zermelo-Fraenkel.
	Lược đồ Cantor	Số lượng tập con của tập hợp có bậc lớn hơn số phần tử – dẫn đến phân biệt $\aleph_0, \aleph_1, \dots$
	Giả thuyết liên tục (Continuum Hypothesis)	Có tồn tại tập hợp có lực lượng giữa \aleph_0 và \aleph_1 không? Được chứng minh là không thể quyết định trong ZFC.
Xác suất rời rạc	Bài toán xác suất sinh nhật	Dù có vẻ nghịch lý, chỉ cần 23 người là có >50% khả năng 2 người trùng ngày sinh.

Môn học	Tên bài toán kinh điển	Mô tả / Giải thích
	Bài toán Monty Hall	Chọn 1 trong 3 cửa, sau khi lộ 1 cửa sai, nên đổi hay giữ? Đổi giúp tăng cơ hội từ 1/3 lên 2/3.
	Xếp ngẫu nhiên mũ	N người đội mũ ngẫu nhiên – xác suất ít nhất 1 người đội đúng mũ là gần 1 – đây là bài toán xáo trộn (derangement).
Lý thuyết đồ thị	Bài toán cầu Königsberg	Euler chứng minh không thể đi qua tất cả các cây cầu 1 lần mà không lặp – khai sinh lý thuyết đồ thị.
	Bài toán người đưa thư Trung Quốc	Tìm chu trình đi qua mọi cạnh ít nhất 1 lần với chi phí thấp nhất – ứng dụng thực tế.
	Bài toán Hamiltonian vs Eulerian	Một đường đi qua mọi đỉnh (Hamilton) và mọi cạnh (Euler) – khác biệt cơ bản trong lý thuyết đồ thị.
	Tô màu đồ thị (Graph Coloring)	Tìm số màu tối thiểu để tô các đỉnh sao cho không có hai đỉnh kề nhau cùng màu.
Lý thuyết số	Bài toán Fermat nhỏ và lớn	Số nguyên n thỏa mãn điều kiện không có nghiệm – nổi tiếng và dẫn đến định lý lớn của Fermat.
	Phân tích số nguyên thành thừa số nguyên tố	Cơ sở của mật mã RSA, là bài toán khó trong máy tính.
	Bài toán Goldbach	Mọi số chẵn > 2 là tổng của hai số nguyên tố? Chưa chứng minh được.
	Kiểm tra số nguyên tố (AKS, Miller-Rabin)	Tìm thuật toán kiểm tra nhanh số nguyên tố – quan trọng trong lý thuyết tính toán và mật mã.
Hình học rời rạc	Đóng gói hình tròn (Circle Packing)	Cách sắp xếp tối ưu các hình tròn trong mặt phẳng – ứng dụng trong vật liệu học.
	Bài toán số Erdős–Szekeres	Mỗi tập n điểm trong mặt phẳng tổng quát đều chứa một tập con 5 điểm tạo thành hình ngũ giác lồi.
	Bài toán bao lồi (Convex Hull)	Cho n điểm, tìm đa giác lồi nhỏ nhất chứa tất cả chúng – cơ bản trong hình học tính toán.
Lý thuyết tính toán	Bài toán Halting (dừng chương trình)	Không tồn tại thuật toán tổng quát kiểm tra xem chương trình có dừng không.
	Bài toán Post Correspondence	Một trong những bài toán đầu tiên chứng minh là không quyết định được.
	Máy Turing	Mô hình trừu tượng nhưng rất mạnh để mô tả thuật toán – nền tảng của lý thuyết tính toán.
Lý thuyết phạm trù	Bài toán giới hạn và đối giới hạn (Limit/Colimit)	Tìm cấu trúc phổ quát trong phạm trù – tổng quát hóa các phép hợp, giao.
	Định lý Yoneda	Mọi đối tượng trong phạm trù có thể được hiểu thông qua các ánh xạ từ nó – cực kỳ quan trọng trong đại số trừu tượng và lý thuyết functor.
	Bài toán tích phủ (Pullback/Pushout)	Tổng quát hóa giao và hợp có điều kiện trong phạm trù – ứng dụng trong logic và topology.

Bảng danh sách các bài toán kinh điển theo môn toán

Môn toán	Tên bài toán	Mô tả ngắn gọn	Liên hệ với TCS	Ứng dụng thực tế
Đại số trừu tượng	Phân loại nhóm hữu hạn (Finite Group Classification)	Phân loại tất cả các nhóm hữu hạn dựa trên bậc và tính chất (cyclic, Abel, không Abel).	Sử dụng lý thuyết nhóm để mô tả cấu trúc đại số, liên quan đến biểu diễn dữ liệu và thuật toán. Là nền tảng cho mật mã học và lý thuyết mã hóa.	Mật mã học (RSA, mã dựa trên nhóm), mã sửa lỗi, phân tích đối xứng trong đồ thị.
	Tìm đồng cấu nhóm (Group Homomorphism Problem)	Tìm tất cả các đồng cấu từ nhóm G sang nhóm H, hoặc xác định xem có đồng cấu nào tồn tại không.	Liên quan đến biểu diễn dữ liệu (ánh xạ cấu trúc) và thuật toán đồ thị (Graph Isomorphism).	Mật mã học, phân tích cấu trúc dữ liệu, thiết kế giao thức an toàn.
	Bài toán từ đẳng cấu (Word Problem for Groups)	Cho một nhóm hữu hạn được sinh bởi tập S, xác định liệu một từ (kết hợp các phần tử trong S) có biểu diễn phần tử đơn vị hay không.	Liên quan đến tính toán trong nhóm và lý thuyết tính toán. Là bài toán không quyết định được trong một số trường hợp.	Kiểm chứng hình thức, mật mã học dựa trên nhóm, phân tích chương trình.
Logic	Thỏa mãn (SAT Problem)	Xác định liệu một công thức logic mệnh đề dạng chuẩn (CNF) có cách	Bài toán NP-complete đầu tiên, thúc đẩy nghiên cứu độ phức	Kiểm chứng phần mềm/phản cứng,

Môn toán	Tên bài toán	Mô tả ngắn gọn	Liên hệ với TCS	Ứng dụng thực tế
		gán giá trị (đúng/sai) để công thức đúng hay không.	tập và thuật toán xấp xỉ. Liên quan đến logic và kiểm chứng hình thức.	suy luận AI, lập kế hoạch.
	Kiểm chứng mô hình (Model Checking)	Xác định liệu một hệ thống (mô hình hóa bằng automata) có thỏa mãn một công thức logic (thường là temporal logic) hay không.	Sử dụng logic để kiểm chứng tính đúng đắn, liên quan đến biểu diễn dữ liệu và thuật toán tối ưu.	Kiểm tra lỗi trong chip, giao thức mạng, hệ thống nhúng.
	Quyết định logic vị từ (Decision Problem for Predicate Logic)	Xác định liệu một công thức logic vị từ có đúng trong mọi mô hình hay không (vấn đề Entscheidungsproblem).	Là bài toán không quyết định được, minh họa giới hạn của logic và tính toán trong TCS.	Kiểm chứng hình thức, thiết kế ngôn ngữ lập trình, suy luận tự động.
Toán rời rạc	Đường đi ngắn nhất (Shortest Path Problem)	Tìm đường đi ngắn nhất từ đỉnh nguồn đến đỉnh đích trong một đồ thị có trọng số.	Ví dụ điển hình về thiết kế thuật toán tối ưu (Dijkstra, Bellman-Ford), liên quan đến toán rời rạc và lý thuyết đồ thị.	Định tuyến mạng, GPS, phân tích mạng xã hội.
	Cây bao trùm tối thiểu (Minimum Spanning Tree)	Tìm cây bao trùm với tổng trọng số nhỏ nhất trong một đồ thị liên thông có trọng số.	Liên quan đến tối ưu tổ hợp và thuật toán (Kruskal, Prim).	Thiết kế mạng cáp quang, phân cụm dữ liệu, logistics.
	Luồng tối đa (Maximum Flow)	Tìm luồng lớn nhất từ nguồn đến đích trong một mạng luồng có dung lượng trên các cạnh.	Liên quan đến thuật toán tối ưu (Ford-Fulkerson) và tối ưu tổ hợp.	Tối ưu hóa băng thông mạng, lập lịch sản xuất, phân phối hàng hóa.
Tổ hợp	Bao phủ tập hợp (Set Cover Problem)	Tìm tập hợp nhỏ nhất các tập con của tập vũ trụ U sao cho hợp của chúng bao phủ U.	Bài toán NP-complete, liên quan đến tối ưu tổ hợp và lý thuyết thông tin.	Tối ưu hóa trạm phát sóng, lập lịch, phân bổ tài nguyên.
	Đếm hoán vị (Permutation Counting)	Tính số cách sắp xếp n đối tượng, có thể có ràng buộc (như tránh hai đối tượng cụ thể đứng cạnh nhau).	Liên quan đến tối ưu tổ hợp và lý thuyết thông tin (đếm cấu hình).	Lập lịch, phân tích dữ liệu, thiết kế thí nghiệm.
	Thiết kế khối (Combinatorial Design)	Xây dựng các cấu trúc tổ hợp (như khối hoặc ma trận) thỏa mãn các ràng buộc, ví dụ: mỗi cặp phần tử xuất hiện đúng k lần.	Liên quan đến lý thuyết mã hóa và tối ưu tổ hợp, tương tự bài toán con heo.	Mã sửa lỗi, thiết kế thí nghiệm khoa học, mật mã học.
Lý thuyết tập hợp	Đếm tập hợp (Cardinality of Sets)	Xác định số đếm của một tập hợp, đặc biệt khi so sánh tập hợp vô hạn (số tự nhiên vs. số thực).	Cung cấp ngôn ngữ cho biểu diễn dữ liệu và logic trong TCS.	Ngữ nghĩa ngôn ngữ lập trình, lý thuyết mô hình, phân tích tính toán.
	Giả thuyết Continuum (Continuum Hypothesis)	Xác định liệu có tập hợp nào có số đếm giữa số tự nhiên (\aleph_0) và số thực (2^{\aleph_0}) hay không.	Liên quan đến logic và lý thuyết mô hình, minh họa giới hạn của lý thuyết tập hợp trong TCS.	Nghiên cứu nền tảng toán học, logic vị từ, giới hạn tính toán.
	Bài toán ánh xạ song ánh (Bijection Problem)	Tìm ánh xạ song ánh giữa hai tập hợp để chứng minh chúng có cùng số đếm.	Liên quan đến biểu diễn dữ liệu (ánh xạ cấu trúc) và logic.	Phân tích cấu trúc dữ liệu, thiết kế kiểu dữ liệu, kiểm chứng hình thức.
Xác suất rời rạc	Phân tích thuật toán ngẫu nhiên (Randomized Algorithm Analysis)	Phân tích hiệu suất của thuật toán ngẫu nhiên, như QuickSort hoặc kiểm tra tính nguyên tố.	Liên quan đến thiết kế thuật toán tối ưu, sử dụng xác suất để đạt hiệu quả kỳ vọng.	Mật mã học, học máy, phân tích dữ liệu lớn.
	Chuỗi Markov (Markov Chain Analysis)	Phân tích hành vi lâu dài của chuỗi Markov, như xác suất trạng thái ổn định hoặc thời gian trộn.	Liên quan đến biểu diễn dữ liệu (mô hình hóa hệ thống động) và lý thuyết thông tin.	Phân tích mạng, mô phỏng hệ thống, học máy (HMM).
	Bất đẳng thức Chernoff (Chernoff Bounds)	Áp dụng bất đẳng thức để phân tích xác suất của các sự kiện ngẫu nhiên trong thuật toán hoặc hệ thống.	Liên quan đến lý thuyết thông tin và thuật toán ngẫu nhiên, tối ưu hóa phân tích.	Phân tích hiệu suất mạng, học máy, kiểm tra độ tin cậy.
Lý thuyết đồ thị	Ghép đôi tối đa (Maximum Matching)	Tìm tập hợp lớn nhất các cạnh trong đồ thị sao cho không có hai cạnh nào chung đỉnh.	Liên quan đến tối ưu tổ hợp và thuật toán (Hungarian, Ford-Fulkerson).	Phân bổ tài nguyên, lập lịch, phân tích mạng xã hội.
	Tô màu đồ thị (Graph Coloring)	Gán màu cho các đỉnh của đồ thị sao cho không có hai đỉnh kề nhau cùng màu, với số màu ít nhất.	Bài toán NP-complete, liên quan đến tối ưu tổ hợp và thuật toán xấp xỉ.	Lập lịch (tránh trùng lịch), phân bổ tần số radio, tối ưu hóa mạng.

Môn toán	Tên bài toán	Mô tả ngắn gọn	Liên hệ với TCS	Ứng dụng thực tế
	Đồng cấu đồ thị (Graph Isomorphism)	Xác định liệu hai đồ thị có đồng cấu (có cùng cấu trúc) hay không.	Liên quan đến đại số trừu tượng (nhóm đối xứng) và thuật toán tối ưu.	Phân tích cấu trúc mạng, nhận dạng mẫu, mật mã học.
Lý thuyết số	Kiểm tra tính nguyên tố (Primality Testing)	Xác định liệu một số n có phải là số nguyên tố hay không.	Liên quan đến thuật toán tối ưu (Miller-Rabin, AKS) và lý thuyết số, nền tảng cho mật mã học.	Mật mã học (RSA, Diffie-Hellman), bảo mật hệ thống.
	Phân tích số (Integer Factorization)	Phân tích một số hợp số thành các thừa số nguyên tố.	Liên quan đến thuật toán tối ưu (Quadratic Sieve, Number Field Sieve) và đại số trừu tượng.	Mật mã học (phá mã RSA), bảo mật giao thức.
	Đồng dư tuyến tính (Linear Congruence)	Giải phương trình đồng dư tuyến tính $ax \equiv b \pmod{m}$ để tìm x.	Liên quan đến lý thuyết số và thuật toán (Euclid mở rộng).	Mật mã học, lập lịch, thiết kế mã hóa.
Hình học rời rạc	Bao lồi (Convex Hull)	Tìm đa giác lồi nhỏ nhất chứa tất cả các điểm trong một tập hợp điểm trên mặt phẳng.	Liên quan đến thuật toán tối ưu (Graham's Scan, $O(n \log n)$).	Đồ họa máy tính, thiết kế vi mạch, phân tích dữ liệu không gian.
	Người bán hàng (Traveling Salesman Problem, TSP)	Tìm hành trình ngắn nhất đi qua tất cả các thành phố và quay về điểm xuất phát.	Bài toán NP-complete, liên quan đến tối ưu tổ hợp và thuật toán xấp xỉ.	Logistics, định tuyến mạng, tối ưu hóa chuỗi cung ứng.
	Định lý Pick (Pick's Theorem)	Tính diện tích của một đa giác trên lưới dựa trên số điểm lưới bên trong và trên biên.	Liên quan đến biểu diễn dữ liệu (mô tả không gian rời rạc) và tổ hợp.	Đồ họa máy tính, phân tích hình học, thiết kế hệ thống nhúng.
Lý thuyết tính toán	Dừng (Halting Problem)	Xác định liệu một chương trình trên máy Turing với đầu vào cụ thể sẽ dừng lại hay chạy mãi mãi.	Bài toán không quyết định được, minh họa giới hạn của logic và tính toán.	Phân tích chương trình, kiểm chứng phần mềm, giới hạn AI.
	Ngôn ngữ hình thức (Language Recognition)	Xác định liệu một ngôn ngữ có được chấp nhận bởi một automata (DFA, PDA) hay không.	Liên quan đến biểu diễn dữ liệu (ngôn ngữ) và lý thuyết tính toán.	Thiết kế trình biên dịch, phân tích cú pháp, xử lý ngôn ngữ tự nhiên.
	Bài toán P vs NP	Xác định liệu mọi bài toán có lời giải kiểm tra trong thời gian đa thức (NP) có thể được giải trong thời gian đa thức (P) hay không.	Là câu hỏi trung tâm của TCS, liên quan đến độ phức tạp và thuật toán tối ưu.	Tối ưu hóa thuật toán, mật mã học, phân tích hiệu suất hệ thống.
Lý thuyết phạm trù	Tìm hàm tử (Functor Construction)	Xây dựng một hàm tử giữa hai phạm trù bảo toàn cấu trúc.	Liên quan đến biểu diễn dữ liệu trừu tượng và suy luận cấu trúc, nền tảng cho ngữ nghĩa lập trình.	Ngữ nghĩa ngôn ngữ lập trình, lý thuyết loại, kiểm chứng hình thức.
	Biến đổi tự nhiên (Natural Transformation)	Tìm một biến đổi tự nhiên giữa hai hàm tử F và G giữa hai phạm trù.	Liên quan đến logic và biểu diễn dữ liệu, giúp tổng quát hóa cấu trúc tính toán.	Lập trình hàm (Haskell), kiểm chứng hệ thống, lý thuyết loại.
	Đối ngẫu phạm trù (Categorical Duality)	Tìm các khái niệm đối ngẫu trong một phạm trù, như sản phẩm và coproduct, để khám phá mối quan hệ.	Liên quan đến biểu diễn dữ liệu và suy luận trừu tượng, hỗ trợ thiết kế hệ thống.	Thiết kế kiểu dữ liệu, kiểm chứng hình thức, tối ưu hóa chương trình.

Giải thích chi tiết cho từng bài toán

Đại số trừu tượng

- Phân loại nhóm hữu hạn:**
 - Mô tả chi tiết:** Bài toán yêu cầu liệt kê tất cả các nhóm hữu hạn theo bậc (số phần tử) và phân loại chúng thành các loại như cyclic, Abel, hoặc không Abel. Định lý phân loại nhóm hữu hạn là một thành tựu lớn, xác định mọi nhóm hữu hạn là tổ hợp của các nhóm đơn giản.
 - Liên hệ với TCS:** Nhóm hữu hạn được dùng trong **biểu diễn dữ liệu** (mô tả hành vi nhóm) và **mật mã học** (như RSA, mã dựa trên nhóm cyclic). Liên quan đến mối quan tâm của bạn về đại số trừu tượng và biểu diễn dữ liệu.
 - Ứng dụng thực tế:** Mật mã học, mã sửa lỗi, phân tích đối xứng trong thuật toán đồ thị.

- **Ví dụ:** Xác định liệu một nhóm bậc 12 là cyclic hay isomorphic với nhóm dihedral D_{12} .
2. **Tìm đồng cấu nhóm:**
- **Mô tả chi tiết:** Cho hai nhóm G và H , tìm tất cả các đồng cấu (ánh xạ bảo toàn cấu trúc) từ G sang H . Bài toán này yêu cầu hiểu cấu trúc nhóm và tính chất đồng cấu.
 - **Liên hệ với TCS:** Đồng cấu nhóm liên quan đến **biểu diễn dữ liệu** (ánh xạ giữa các cấu trúc) và thuật toán, như trong bài toán Graph Isomorphism. Phù hợp với mối quan tâm của bạn về đại số trừu tượng.
 - **Ứng dụng thực tế:** Mật mã học (thiết kế giao thức), phân tích cấu trúc dữ liệu, kiểm chứng hình thức.
 - **Ví dụ:** Tìm các đồng cấu từ nhóm Z_4 sang nhóm S_3 .
3. **Bài toán từ đẳng cấu:**
- **Mô tả chi tiết:** Cho một nhóm hữu hạn được sinh bởi tập S , xác định liệu một từ (chuỗi các phần tử trong S) có biểu diễn phần tử đơn vị hay không. Bài toán này không quyết định được trong một số nhóm.
 - **Liên hệ với TCS:** Liên quan đến **lý thuyết tính toán** và **kiểm chứng hình thức**, vì nó kiểm tra tính đúng đắn của các phép tính trong nhóm. Phù hợp với mối quan tâm của bạn về đại số trừu tượng.
 - **Ứng dụng thực tế:** Mật mã học dựa trên nhóm, kiểm chứng chương trình, phân tích cấu trúc tính toán.
 - **Ví dụ:** Trong nhóm tự do với hai sinh, xác định liệu từ $abab^{-1}a^{-1}$ có là phần tử đơn vị hay không.

Logic

4. **Thỏa mãn (SAT Problem):**
- **Mô tả chi tiết:** Cho một công thức logic mệnh đề dạng chuẩn (CNF), tìm gán giá trị (đúng/sai) cho các biến để công thức đúng. SAT là bài toán NP-complete đầu tiên (Định lý Cook-Levin).
 - **Liên hệ với TCS:** Là trung tâm của nghiên cứu độ phức tạp, liên quan đến **logic** và **thiết kế thuật toán tối ưu**. Các trình giải SAT (Z3, MiniSAT) được dùng trong kiểm chứng hình thức.
 - **Ứng dụng thực tế:** Kiểm tra phần mềm/phần cứng, lập kế hoạch, suy luận trong AI.
 - **Ví dụ:** Kiểm tra tính thỏa mãn của $(A \vee \neg B) \wedge (\neg A \vee C) \wedge (B \vee \neg C)$.
5. **Kiểm chứng mô hình:**
- **Mô tả chi tiết:** Cho một hệ thống (mô hình hóa bằng automata) và một công thức logic (thường là temporal logic như LTL, CTL), xác định liệu hệ thống thỏa mãn công thức đó hay không.
 - **Liên hệ với TCS:** Sử dụng **logic** để kiểm chứng tính đúng đắn, liên quan đến biểu diễn dữ liệu (mô hình hóa hệ thống) và thuật toán tối ưu (giảm không gian trạng thái).
 - **Ứng dụng thực tế:** Kiểm tra lỗi trong chip, giao thức mạng, hệ thống nhúng.
 - **Ví dụ:** Kiểm tra xem một giao thức mạng có luôn đảm bảo không mất gói tin hay không.
6. **Quyết định logic vị từ:**
- **Mô tả chi tiết:** Xác định liệu một công thức logic vị từ có đúng trong mọi mô hình hay không. Turing và Church chứng minh bài toán này không quyết định được (Entscheidungsproblem).
 - **Liên hệ với TCS:** Minh họa giới hạn của **logic** và **tính toán**, liên quan đến kiểm chứng hình thức và lý thuyết tính toán.
 - **Ứng dụng thực tế:** Kiểm chứng hình thức, thiết kế ngôn ngữ lập trình, suy luận tự động.
 - **Ví dụ:** Kiểm tra xem công thức $\forall x \exists y (x + y = 0)$ có đúng trong mọi trường hay không.

Toán rời rạc

7. **Đường đi ngắn nhất:**
- **Mô tả chi tiết:** Tìm đường đi ngắn nhất từ đỉnh nguồn đến đỉnh đích trong đồ thị có trọng số, sử dụng thuật toán như Dijkstra ($O(V^2)$), Bellman-Ford ($O(VE)$), hoặc A* (heuristic).
 - **Liên hệ với TCS:** Là ví dụ kinh điển về **thiết kế thuật toán tối ưu**, liên quan đến toán rời rạc và lý thuyết đồ thị. Phù hợp với mối quan tâm của bạn về tối ưu hóa space-time.
 - **Ứng dụng thực tế:** Định tuyến mạng, GPS, phân tích mạng xã hội.
 - **Ví dụ:** Tìm đường đi ngắn nhất từ thành phố A đến B trong mạng giao thông.
8. **Cây bao trùm tối thiểu:**
- **Mô tả chi tiết:** Tìm cây bao trùm với tổng trọng số nhỏ nhất trong đồ thị liên thông, sử dụng thuật toán Kruskal hoặc Prim ($O(E \log V)$).
 - **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và thuật toán, là bài toán cơ bản trong toán rời rạc và lý thuyết đồ thị.
 - **Ứng dụng thực tế:** Thiết kế mạng cáp quang, phân cụm dữ liệu, logistics.
 - **Ví dụ:** Tìm cách nối các máy tính trong văn phòng với tổng chiều dài dây nhỏ nhất.
9. **Luồng tối đa:**
- **Mô tả chi tiết:** Tìm luồng lớn nhất từ nguồn đến đích trong một mạng luồng có dung lượng, sử dụng thuật toán Ford-Fulkerson hoặc Edmonds-Karp ($O(VE^2)$).
 - **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và thuật toán tối ưu, là bài toán kinh điển trong toán rời rạc.
 - **Ứng dụng thực tế:** Tối ưu hóa băng thông mạng, lập lịch sản xuất, phân phối hàng hóa.

- **Ví dụ:** Tính luồng tối đa của nước qua một hệ thống ống dẫn.

Tổ hợp

10. Bao phủ tập hợp:

- **Mô tả chi tiết:** Tìm tập hợp nhỏ nhất các tập con của tập vũ trụ U sao cho hợp của chúng bao phủ U . Là bài toán NP-complete, thường giải bằng thuật toán tham lam.
- **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và **lý thuyết thông tin**, tương tự bài toán con heo trong việc tối ưu hóa lựa chọn.
- **Ứng dụng thực tế:** Tối ưu hóa trạm phát sóng, lập lịch, phân bổ tài nguyên.
- **Ví dụ:** Tìm số lượng tối thiểu các trạm cứu hỏa để bao phủ mọi khu vực.

11. Đếm hoán vị:

- **Mô tả chi tiết:** Tính số cách sắp xếp n đối tượng, có thể có ràng buộc (như A và B không đứng cạnh nhau). Sử dụng nguyên lý bao hàm-loại trừ hoặc hàm sinh.
- **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và **lý thuyết thông tin** (đếm cấu hình), phù hợp với mối quan tâm của bạn về tối ưu hóa.
- **Ứng dụng thực tế:** Lập lịch, phân tích dữ liệu, thiết kế thí nghiệm.
- **Ví dụ:** Tính số cách xếp 5 người sao cho A và B không đứng cạnh nhau.

12. Thiết kế khối:

- **Mô tả chi tiết:** Xây dựng các cấu trúc tổ hợp (như khối hoặc ma trận) thỏa mãn các ràng buộc, ví dụ: mỗi cặp phần tử xuất hiện đúng k lần trong các tập con.
- **Liên hệ với TCS:** Liên quan đến **lý thuyết mã hóa** và **tối ưu tổ hợp**, tương tự bài toán con heo trong việc thiết kế thí nghiệm tối ưu.
- **Ứng dụng thực tế:** Mã sửa lỗi, thiết kế thí nghiệm khoa học, mật mã học.
- **Ví dụ:** Thiết kế một khối với 7 phần tử, mỗi tập con có 3 phần tử, và mỗi cặp xuất hiện đúng một lần.

Lý thuyết tập hợp

13. Đếm tập hợp:

- **Mô tả chi tiết:** Xác định số đếm của tập hợp, đặc biệt khi so sánh tập hợp vô hạn. Ví dụ, tập số tự nhiên có số đếm \aleph_0 , tập số thực có 2^{\aleph_0} .
- **Liên hệ với TCS:** Cung cấp ngôn ngữ cho **biểu diễn dữ liệu** và **logic**, giúp định nghĩa kiểu dữ liệu và ngôn ngữ lập trình.
- **Ứng dụng thực tế:** Ngữ nghĩa ngôn ngữ lập trình, lý thuyết mô hình, phân tích tính toán.
- **Ví dụ:** Chứng minh tập số thực có số đếm lớn hơn tập số tự nhiên.

14. Giả thuyết Continuum:

- **Mô tả chi tiết:** Xác định liệu có tập hợp nào có số đếm giữa \aleph_0 (số tự nhiên) và 2^{\aleph_0} (số thực). Gödel và Cohen chứng minh bài toán này độc lập với ZFC.
- **Liên hệ với TCS:** Liên quan đến **logic** và lý thuyết mô hình, minh họa giới hạn của lý thuyết tập hợp.
- **Ứng dụng thực tế:** Nghiên cứu nền tảng toán học, logic vị từ, giới hạn tính toán.
- **Ví dụ:** Nghiên cứu các hệ quả của giả thuyết Continuum trong ZFC.

15. Ánh xạ song ánh:

- **Mô tả chi tiết:** Tìm ánh xạ song ánh giữa hai tập hợp để chứng minh chúng có cùng số đếm. Ví dụ, chứng minh tập số chẵn và số tự nhiên có cùng số đếm.
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** (ánh xạ cấu trúc) và logic, hỗ trợ phân tích cấu trúc tính toán.
- **Ứng dụng thực tế:** Phân tích cấu trúc dữ liệu, thiết kế kiểu dữ liệu, kiểm chứng hình thức.
- **Ví dụ:** Tìm ánh xạ song ánh giữa tập số tự nhiên và tập số nguyên.

Xác suất rời rạc

16. Phân tích thuật toán ngẫu nhiên:

- **Mô tả chi tiết:** Phân tích hiệu suất của thuật toán ngẫu nhiên, như QuickSort ngẫu nhiên ($O(n \log n)$ kỳ vọng) hoặc Miller-Rabin cho kiểm tra tính nguyên tố.
- **Liên hệ với TCS:** Liên quan đến **thiết kế thuật toán tối ưu** và **lý thuyết thông tin**, vì xác suất cải thiện hiệu quả kỳ vọng.
- **Ứng dụng thực tế:** Mật mã học, học máy, phân tích dữ liệu lớn.
- **Ví dụ:** Tính kỳ vọng số phép so sánh trong QuickSort ngẫu nhiên.

17. Chuỗi Markov:

- **Mô tả chi tiết:** Phân tích hành vi lâu dài của chuỗi Markov, như xác suất trạng thái ổn định hoặc thời gian trộn (mixing time).
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** (mô hình hóa hệ thống động) và **lý thuyết thông tin** (entropy của trạng thái).
- **Ứng dụng thực tế:** Phân tích mạng, mô phỏng hệ thống, học máy (Hidden Markov Models).
- **Ví dụ:** Tính xác suất một người dùng ở lại trang web sau nhiều bước chuyển đổi.

18. Bất đẳng thức Chernoff:

- **Mô tả chi tiết:** Áp dụng bất đẳng thức Chernoff để phân tích xác suất của các sự kiện ngẫu nhiên, như sai lệch của tổng biến ngẫu nhiên từ kỳ vọng.
- **Liên hệ với TCS:** Liên quan đến **lý thuyết thông tin** và thuật toán ngẫu nhiên, giúp phân tích hiệu suất và độ tin cậy.
- **Ứng dụng thực tế:** Phân tích hiệu suất mạng, học máy, kiểm tra độ tin cậy hệ thống.
- **Ví dụ:** Ước lượng xác suất một thuật toán ngẫu nhiên sai lệch lớn từ kết quả mong muốn.

Lý thuyết đồ thị

19. Ghép đôi tối đa:

- **Mô tả chi tiết:** Tìm tập hợp lớn nhất các cạnh trong đồ thị sao cho không có hai cạnh nào chung đỉnh, sử dụng thuật toán Hungarian hoặc Ford-Fulkerson.
- **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và thuật toán tối ưu, là bài toán kinh điển trong lý thuyết đồ thị.
- **Ứng dụng thực tế:** Phân bổ tài nguyên (ghép nhân viên với công việc), lập lịch, phân tích mạng.
- **Ví dụ:** Ghép đôi tối đa các ứng viên với vị trí công việc.

20. Tô màu đồ thị:

- **Mô tả chi tiết:** Gán màu cho các đỉnh sao cho không có hai đỉnh kề nhau cùng màu, với số màu ít nhất (chromatic number). Là bài toán NP-complete.
- **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và thuật toán xấp xỉ, phù hợp với mối quan tâm của bạn về tối ưu hóa.
- **Ứng dụng thực tế:** Lập lịch (tránh trùng lịch), phân bổ tần số radio, tối ưu hóa mạng.
- **Ví dụ:** Tô màu các phòng học để tránh trùng lịch học.

21. Đồng cấu đồ thị:

- **Mô tả chi tiết:** Xác định liệu hai đồ thị có đồng cấu (có cùng cấu trúc) hay không. Bài toán này có độ phức tạp chưa được xác định rõ (không biết là P hay NP-complete).
- **Liên hệ với TCS:** Liên quan đến **đại số trừu tượng** (nhóm đối xứng) và **thuật toán tối ưu**, phù hợp với mối quan tâm của bạn về đại số.
- **Ứng dụng thực tế:** Phân tích cấu trúc mạng, nhận dạng mẫu, mật mã học.
- **Ví dụ:** Kiểm tra xem hai đồ thị mạng xã hội có đồng cấu hay không.

Lý thuyết số

22. Kiểm tra tính nguyên tố:

- **Mô tả chi tiết:** Xác định liệu một số n có phải là số nguyên tố hay không, sử dụng thuật toán Miller-Rabin (ngẫu nhiên) hoặc AKS (xác định).
- **Liên hệ với TCS:** Liên quan đến **thuật toán tối ưu** và **lý thuyết số**, là nền tảng cho mật mã học.
- **Ứng dụng thực tế:** Mật mã học (RSA, Diffie-Hellman), bảo mật hệ thống.
- **Ví dụ:** Kiểm tra xem 101 là số nguyên tố hay không.

23. Phân tích số:

- **Mô tả chi tiết:** Phân tích một số hợp số thành các thừa số nguyên tố, sử dụng thuật toán như Quadratic Sieve hoặc Number Field Sieve.
- **Liên hệ với TCS:** Liên quan đến **thuật toán tối ưu** và **đại số trừu tượng**, vì nó dựa trên cấu trúc trường hữu hạn.
- **Ứng dụng thực tế:** Mật mã học (phá mã RSA), bảo mật giao thức.
- **Ví dụ:** Phân tích 100 thành $2^2 * 5^2$.

24. Đồng dư tuyến tính:

- **Mô tả chi tiết:** Giải phương trình $ax \equiv b \pmod{m}$ để tìm x , sử dụng thuật toán Euclid mở rộng để tính GCD.
- **Liên hệ với TCS:** Liên quan đến **lý thuyết số** và **thuật toán tối ưu**, là nền tảng cho mật mã học và lập lịch.
- **Ứng dụng thực tế:** Mật mã học, lập lịch, thiết kế mã hóa.
- **Ví dụ:** Giải $3x \equiv 2 \pmod{7}$.

Hình học rời rạc

25. Bao lồi:

- **Mô tả chi tiết:** Tìm đa giác lồi nhỏ nhất chứa tất cả các điểm trong một tập hợp điểm trên mặt phẳng, sử dụng thuật toán Graham's Scan hoặc Andrew's Monotone Chain ($O(n \log n)$).
- **Liên hệ với TCS:** Liên quan đến **thuật toán tối ưu** và **biểu diễn dữ liệu** (mô tả không gian rời rạc).
- **Ứng dụng thực tế:** Đồ họa máy tính, thiết kế vi mạch, phân tích dữ liệu không gian.
- **Ví dụ:** Tìm bao lồi của 10 điểm trên mặt phẳng.

26. Người bán hàng (TSP):

- **Mô tả chi tiết:** Tìm hành trình ngắn nhất đi qua tất cả các thành phố và quay về điểm xuất phát. Là bài toán NP-complete, thường giải bằng thuật toán xấp xỉ hoặc heuristic.
- **Liên hệ với TCS:** Liên quan đến **tối ưu tổ hợp** và **thuật toán tối ưu**, phù hợp với mối quan tâm của bạn về tối ưu hóa.

- **Ứng dụng thực tế:** Logistics, định tuyến mạng, tối ưu hóa chuỗi cung ứng.
- **Ví dụ:** Tìm hành trình ngắn nhất qua 5 thành phố.

27. Định lý Pick:

- **Mô tả chi tiết:** Tính diện tích của một đa giác trên lưới dựa trên số điểm lưới bên trong (I) và trên biên (B): Diện tích = $I + B/2 - 1$.
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** (mô tả không gian rời rạc) và **tổ hợp**, hỗ trợ phân tích cấu trúc hình học.
- **Ứng dụng thực tế:** Đồ họa máy tính, phân tích hình học, thiết kế hệ thống nhúng.
- **Ví dụ:** Tính diện tích của một đa giác với 10 điểm bên trong và 8 điểm trên biên.

Lý thuyết tính toán

28. Dừng (Halting Problem):

- **Mô tả chi tiết:** Xác định liệu một chương trình trên máy Turing với đầu vào cụ thể sẽ dừng lại hay chạy mãi mãi. Turing chứng minh bài toán này không quyết định được.
- **Liên hệ với TCS:** Minh họa giới hạn của **logic** và **tính toán**, là bài toán kinh điển trong lý thuyết tính toán.
- **Ứng dụng thực tế:** Phân tích chương trình, kiểm chứng phần mềm, giới hạn AI.
- **Ví dụ:** Kiểm tra xem một chương trình có dừng lại trên mọi đầu vào hay không.

29. Ngôn ngữ hình thức:

- **Mô tả chi tiết:** Xác định liệu một ngôn ngữ có được chấp nhận bởi một automata (DFA, PDA, hoặc máy Turing) hay không.
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** (ngôn ngữ) và lý thuyết tính toán, là nền tảng cho trình biên dịch.
- **Ứng dụng thực tế:** Thiết kế trình biên dịch, phân tích cú pháp, xử lý ngôn ngữ tự nhiên.
- **Ví dụ:** Kiểm tra xem ngôn ngữ $\{a^n b^n\}$ có được chấp nhận bởi PDA hay không.

30. P vs NP:

- **Mô tả chi tiết:** Xác định liệu mọi bài toán có lời giải kiểm tra trong thời gian đa thức (NP) có thể được giải trong thời gian đa thức (P) hay không. Là câu hỏi mở lớn nhất trong TCS.
- **Liên hệ với TCS:** Liên quan đến **độ phức tạp** và **thuật toán tối ưu**, ảnh hưởng đến mọi khía cạnh của TCS.
- **Ứng dụng thực tế:** Tối ưu hóa thuật toán, mật mã học, phân tích hiệu suất hệ thống.
- **Ví dụ:** Nghiên cứu liệu SAT có thể được giải trong thời gian đa thức hay không.

Lý thuyết phạm trù

31. Tìm hàm tử:

- **Mô tả chi tiết:** Xây dựng một hàm tử giữa hai phạm trù bảo toàn cấu trúc, như từ phạm trù tập hợp sang phạm trù nhóm.
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** trừu tượng và **suy luận cấu trúc**, là nền tảng cho ngữ nghĩa lập trình.
- **Ứng dụng thực tế:** Ngữ nghĩa ngôn ngữ lập trình, lý thuyết loại, kiểm chứng hình thức.
- **Ví dụ:** Xây dựng hàm tử từ phạm trù tập hợp sang phạm trù nhóm cyclic.

32. Biến đổi tự nhiên:

- **Mô tả chi tiết:** Tìm một biến đổi tự nhiên giữa hai hàm tử F và G, bảo toàn tính chất của phạm trù.
- **Liên hệ với TCS:** Liên quan đến **logic** và **biểu diễn dữ liệu**, giúp tổng quát hóa cấu trúc tính toán trong lập trình hàm.
- **Ứng dụng thực tế:** Lập trình hàm (Haskell), kiểm chứng hệ thống, lý thuyết loại.
- **Ví dụ:** Tìm biến đổi tự nhiên giữa hai hàm tử trong phạm trù không gian tô pô.

33. Đối ngẫu phạm trù:

- **Mô tả chi tiết:** Tìm các khái niệm đối ngẫu trong một phạm trù, như sản phẩm (product) và coproduct, để khám phá mối quan hệ giữa các cấu trúc.
- **Liên hệ với TCS:** Liên quan đến **biểu diễn dữ liệu** và **suy luận trừu tượng**, hỗ trợ thiết kế hệ thống và kiểm chứng.
- **Ứng dụng thực tế:** Thiết kế kiểu dữ liệu, kiểm chứng hình thức, tối ưu hóa chương trình.
- **Ví dụ:** Nghiên cứu đối ngẫu giữa sản phẩm và coproduct trong phạm trù tập hợp.

Liên hệ với các mối quan tâm của bạn

Các bài toán trên được chọn để liên đới chặt chẽ với các mối quan tâm của bạn:

- **Biểu diễn dữ liệu:** Các bài toán như **Group Homomorphism**, **Cardinality of Sets**, **Language Recognition**, **Functor Construction**, và **Pick's Theorem** liên quan đến việc ánh xạ, mô tả, và biểu diễn dữ liệu trừu tượng hoặc rời rạc, phù hợp với mong muốn của bạn về biểu diễn không dựa vào không gian metric.
- **Thiết kế thuật toán tối ưu:** **Shortest Path**, **Minimum Spanning Tree**, **Maximum Flow**, **Primality Testing**, **Convex Hull**, và **Randomized Algorithm Analysis** là các ví dụ kinh điển về tối ưu hóa độ phức tạp space-time.
- **Lý thuyết thông tin:** **Set Cover**, **Combinatorial Design**, **Markov Chain Analysis**, và **Chernoff Bounds** sử dụng các khái niệm entropy và xác suất, tương tự bài toán con heo trong việc tối ưu hóa thông tin.

- Đại số trừu tượng: **Finite Group Classification**, **Group Homomorphism**, **Word Problem**, **Integer Factorization**, và **Graph Isomorphism** sử dụng cấu trúc nhóm, trường, hoặc đối xứng, phù hợp với mối quan tâm của bạn về hành vi nhóm.
 - Tối ưu tổ hợp: **Set Cover**, **Permutation Counting**, **Maximum Matching**, **Graph Coloring**, **TSP**, và **Maximum Flow** là các bài toán tối ưu tổ hợp, tương tự bài toán con heo trong việc tìm chiến lược tối ưu.
-