Lec00-intro

KT Ch. 1

Sunday, August 23, 2026 2:23 PM

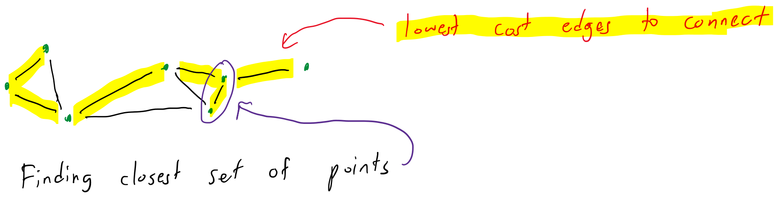
02-613 / 15-650 / 15-351

Objectives: to study general computational problems & their algorithms, with a focus on the principles used to design those algorithms.

Learning goals:

1. Design algorithms using common techniques
2. Prove worst-case runtime
3. Prove a problem is NP-hard
4. Verify validity of given solutions ← especially AI-generated

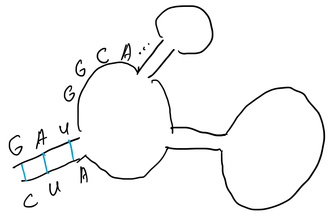
Example 1: Low-cost network design



Example 2: Finding closest set of points

Example 3: RNA folding

GAUGGCAAA UGCAAAGGCCU ....



Example 4: Genomic alignment

ACCGGTTAA
ACGATTCAA

ACCGGTT-AA
| | | | | ||
A-CGATTCAA

General Techniques

Book: Algorithm Design by Jon Kleinberg & Éva Tardos

- Ch. 4: Greedy — easy local optimization at every step
- Ch. 5: Divide & Conquer — break problem into subproblems
- Ch. 6: Dynamic Programming — overlapping subproblems
- Ch. 7: Network Flow — transportation networks → general problem
- 11.6–11.7: Linear & integer programming — constraint satisfaction

Analysis of Algorithms

- Correctness — alg always returns the right answer
- Implementation — what data structures are needed
- Worst-case runtime — speed guarantees
- Computational complexity — prove no alg can do better

Meta-analysis: Determine if an argument of this type is correct.

Course Design:

50% exam
40% partner quizzes
10% tutorials

3 modules

1 exam / module (evenings and/or final)
2 quizzes / module (in lecture)
↑ will include problems from ungraded exercises

Lecture participation

Optional but encouraged.

Vote: Undergrad / CBD Undergrad / Stats Masters Other

Vote: What's more exciting?

Designing algorithms Proving theorems Writing software Pipetting