26 March 2025

Prelim 2 Topics

Plan

* High-Level Overview
* Announcements
* Topics Review

4820 So Far

Prelim 1

* Graph Algorithms
* Greedy Algorithms
* Dynamic Programming

4820 So Far

Prelim 1

* Graph Algorithms
* Greedy Algorithms
* Dynamic Programming

Prelim 2

* Stable Matching
* Network Flow
* FFT & Mathematical Algs
* P vs. NP

Exam Expectations

* Practice Exam Questions are representative!

Exam Expectations

* Practice Exam Questions are representative!

* Lots of topics ⟹ Lots of questions

  ↳ Plan your time accordingly!

  ↳ Look at how many points questions are worth!

Announcements

* HW6 due
  - ↳ Solutions Released this afternoon.

* Prelim 2
  - ↳ Tomorrow 7:30 – 9p.
  - ↳ See Ed for Room Assignments (same as last time)

* Ed Shutdown ~24 hrs before exam
  - ↳ Reopens after Break

* Lecture on Friday!

  (And Enjoy Spring Break!)

Topics

* Stable Matching
* Network Flow
* FFT & Mathematical Algs
* P vs. NP

Stable Matching (2 Lectures)

Given: n Residencies R

n Doctors D

↳ Each Residency has preferences over D

Each Doctor has preferences over R

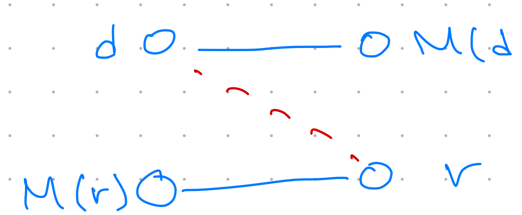
Stable Matching (2 Lectures)

<u>Given</u>. n Residencies R

n Doctors D

↳ Each Residency has preferences over D
Each Doctor has preferences over R

<u>Goal</u>. Find <u>stable perfect matching</u> M
between R & D



<u>Stable</u> $\forall d \in D \quad \forall r \in R$

- d prefers $M(d)$ to r OR
- r prefers $M(r)$ to d

Stable Matching (2 Lectures)

Theorem (Gale & Shapley)

For every instance, there exists a stable perfect matching between R & D.

Stable Matching (2 Lectures)

Theorem (Gale & Shapley)

For every instance, there exists a stable perfect matching between R & D.

Pf. Constructive via Gale-Shapley Algo.

- Residencies make offers to Doctors
- Doctors accept if offer is better than current match.

iterative

Network Flow (6 Lectures)

Given. Flow Network $G = (V, E)$

- source $s \in V$
- sink $t \in V$
- capacities $c : E \to \mathbb{N}$.

Network Flow (6 Lectures)

Given. Flow Network $G = (V, E)$

- source $s \in V$
- sink $t \in V$
- capacities $c : E \to \mathbb{N}$.

Goal.

* Find Maximum Flow $f : E \to \mathbb{N}$.

  s.t. CAPACITY

  & CONSERVATION constraints.

* Find Minimum st. Cut $S \subseteq V$.

Network Flow (6 Lectures)

Max flow = min cut.

Pf. Analysis of Ford-Fulkerson Algorithm!

Network Flow (6 Lectures)

Max flow = min cut.

Pf. Analysis of Ford-Fulkerson Algorithm!

Ford-Fulkerson:

- Compute Residual graph G^f
- Find augmenting st-path & push flow

iterate

Network Flow (6 Lectures)

Reductions to Flow give algorithms!

* In Bipartite Graphs!

* Applications

Network Flow (6 Lectures)

Reductions to Flow give algorithms!

* In Bipartite Graphs!
  - Max Matching
  - Min Vertex Cover (König's Theorem)

* Applications
  - Baseball Elimination
  - Cake Distribution
  - Flow Gadgets

FFT & Mathematical Algs (3 lectures)

↳ No linear systems

<u>Motivation</u>. Given two polynomials p, q compute their product efficiently.

FFT & Mathematical Algs (3 lectures)

↳ No linear systems

Motivation. Given two polynomials p, q compute their product efficiently.

Key observation

* Two equivalent representations of polynomials

  ↳ Coefficient Repr. $\quad p(x) = \sum_{i=0}^{n-1} p_i \cdot x^i$

  ↳ Point-Value Repr.

$$\hat{p} = p(\omega_n)\, p(\omega_n^2) \cdots p(\omega_n^n)$$

FFT & Mathematical Algs (3 lectures)

<u>Given</u> polynomial $p = p_{n-1} p_{n-2} \cdots p_1 p_0$ in coefficient representation

<u>Return</u> evaluation $\hat{p} = \hat{p}(\omega_n)\, \hat{p}(\omega_n^2) \cdots \hat{p}(\omega_n^n)$ in point-value representation on the n^{th} roots of unity.

$$\omega_n^n = 1.$$

$$\omega_n^2 = \omega_{n/2}$$

<u>Pf.</u> $(\omega_n^2)^{n/2} = \omega_n^n = 1$

FFT & Mathematical Algs (3 lectures)

Given polynomial $p = p_{n-1} p_{n-2} \cdots p_1 p_0$
in coefficient representation

Return evaluation $\hat{p} = \hat{p}(\omega_n)\ \hat{p}(\omega_n^2)\ \cdots\ \hat{p}(\omega_n^n)$
in point-value representation
on the n^{th} roots of unity.

Fast Fourier Transform

* Recursive Strategy

$$p(x) = p_{even}(x^2) + x \cdot p_{odd}(x^2)$$

where $p_{even}(y) = \sum_{i=0}^{n/2-1} p_{2i} \cdot y^i$ $\qquad p_{odd}(y) = \sum_{i=0}^{n/2-1} p_{2i+1} \cdot y^i$

FFT & Mathematical Algs (3 lectures)

Fast Fourier Transform

* 2 recursive calls
* degree drops by factor 2 in each call

$$T(n) \leq 2 \cdot T(n/2) + O(n)$$

$$\leq O(n \log n) \text{ basic operations}$$

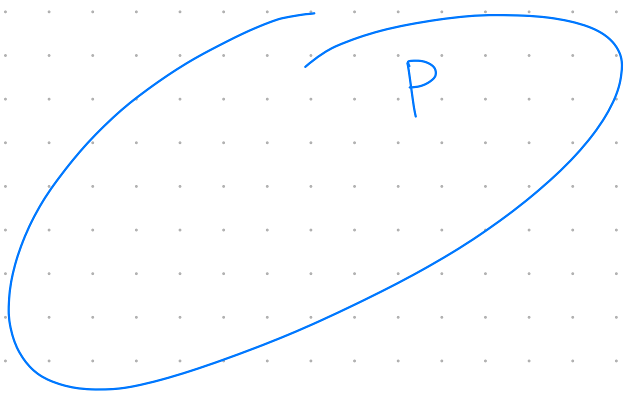
Applications

* Convolution
* Probability Computations

NP - Hardness (4 Lectures)

Motivating Question

Why don't we have efficient algorithms for certain problems?

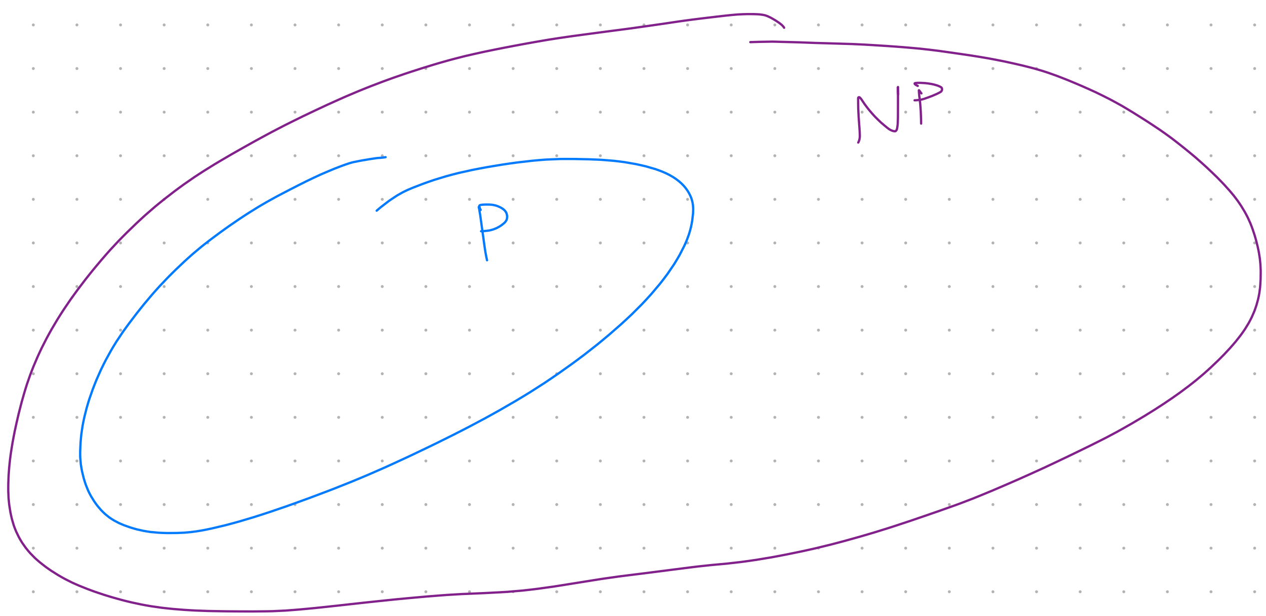


P: problems solvable in polynomial time

NP - Hardness (4 Lectures)

Motivating Question

Why don't we have efficient algorithms for certain problems?

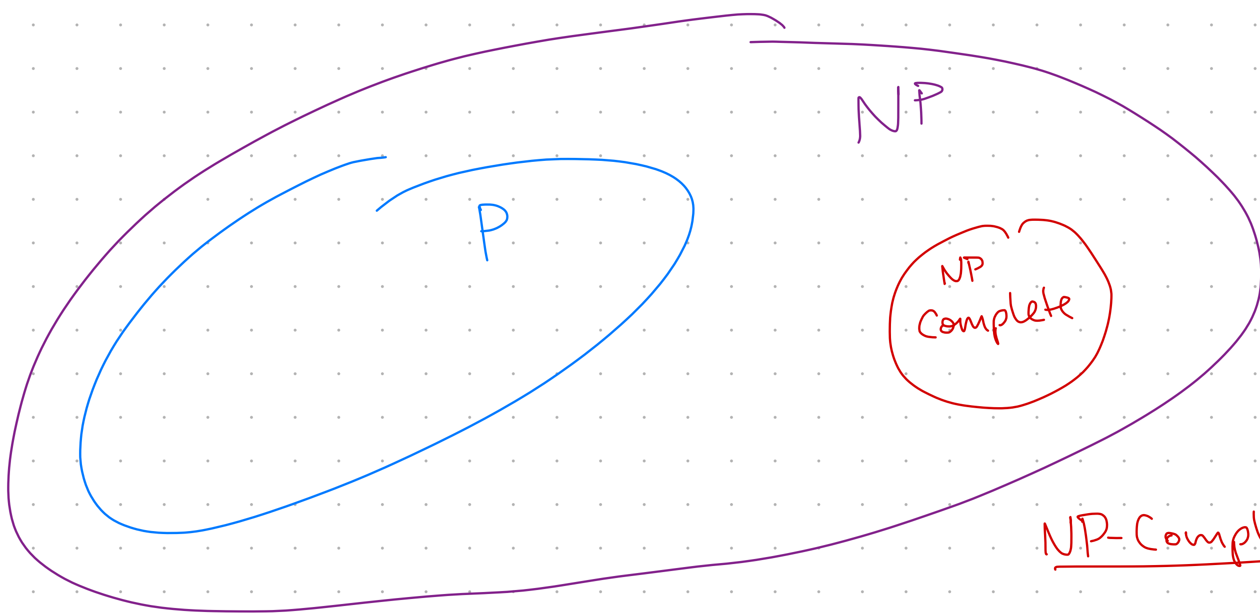


NP = problems verifiable in polynomial time

NP - Hardness (4 Lectures)

Motivating Question

Why don't we have efficient algorithms for certain problems?



NP-Complete ⊆ NP

* Every problem in NP reduces to problem in polynomial time (NP - Hard)

NP-Hardness (4 Lectures)

If $P \neq NP$, then NP-Hard problems cannot be solved in polynomial time.

NP - Hardness (4 Lectures)

If $P \neq NP$, then NP-Hard problems cannot be solved in polynomial time.

NP - Hard Problems

* SAT, 3SAT
* VERTEX COVER, INDEPENDENT SET, CLIQUE
* HAMILTONIAN PATH

Topics

* Stable Matching
* Network Flow
* FFT & Mathematical Algs
* P vs. NP

Questions?

Good Luck!