Learning Goals

- Describe P and NP informally, and why these ideas are important
- Define NP
- Prove a problem is in NP [NP1]

Announcements

CS Seminar on Friday, alum Corey Sch

Reattempts → Office Hours!

Other ET Questions

Can quantum computers solve NP problems?

Industry applications of NP problems?

SAT → satisfiability

NP-Hard → Later!

Array vs hashmap to store Dynamic Progr. subproblems?

go/meetCorey

Middlebury College

Computer Science Seminar

Alumni Tech and Career Talk

Corey Scheinfeld

Software Engineer for Tradeweb

In this talk, Corey will discuss his experience transitioning from studying computer science at Middlebury to working as a software engineer at Tradeweb. He will describe the kinds of systems he works on, and the various pieces of the software development lifecycle that support high-throughput, market central financial platforms. The presentation will connect elements of his undergraduate coursework to their application in production environments, and highlight how a liberal arts background has been critical to his early career in technology. He will also discuss how AI has reshaped the industry, and what tools like Claude and Cursor have changed in the day to day life of a software engineer.

Bio: *Corey Scheinfeld grew up in New Rochelle, New York and attended Middlebury College from Fall, 2018 to Spring, 2022 and graduated with a degree in Computer Science. He has since worked as a software engineer at Tradeweb, an electronic trading marketplace for fixed income, derivatives, ETFs, and equity markets, where he also interned as a student at Middlebury. He works on their institutional technology platform, building scalable trading systems and infrastructure, particularly Node.js based microservices, operating in performance-sensitive environments. In this capacity, he leads a team of engineers on the development of various fixed income trading protocols. Outside of work, Corey enjoys skiing, hiking, and returning to Vermont as often as he can.*

Friday, March 6, 2026 @ 12:20 p.m.

75 Shannon St 102

Pizza will be served

Warm-Ups

What types of problems are in P?

What types of problems are in NP?

If P=NP, would the world change?

Types of Problems

? $1 million

Easy

(Polynomial time)
$O(n)$, $O(n^2)$
$O(n^3)$

- Search
- Sort
- Multiplication
- Greedy Scheduling
- Closest Pts
- Matrix Mult.

Fast algorithm

Quantum Computers

Puzzles

Crosswords $(n \times n)$
Sudoku $(n \times n)$
Delivery Rts
n prime?
Factoring large #
MWIS general graph ≥ 1000

Easy to check
Might be hard to solve

Hard

Chess:
What is next best move?

Can mathematically characterize Easy / Puzzle

P and NP

P (Polynomial Time)

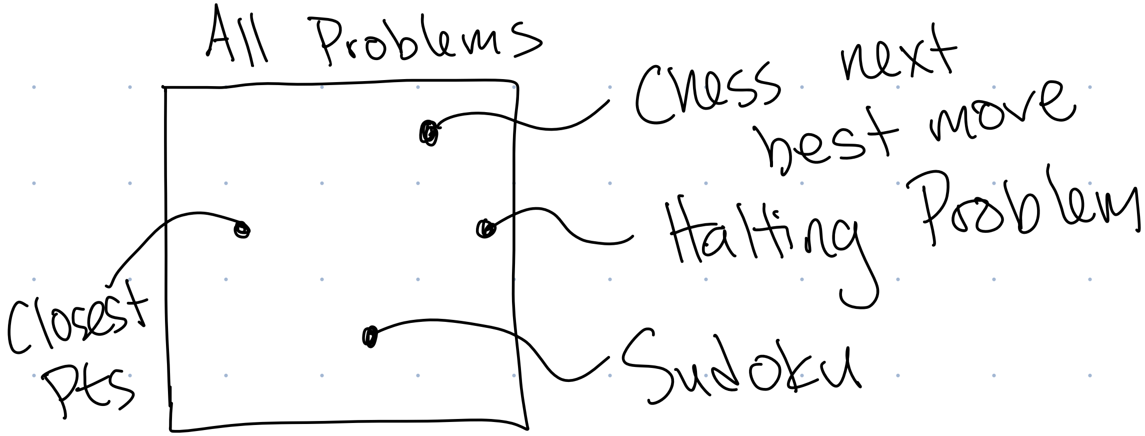
Informal: A problem is in P if it can be solved in polynomial time

NP (Non-deterministic Polynomial Time)

Informal: A problem is in NP if a solution can be checked in polynomial time.

Polynomial Time ??

- Runs in time $O(n^c)$ where n is the size of input and c is a constant. $O(poly(n))$



Which picture is correct?

A B C D

NP problems are YES-NO: $Q(x) = \begin{cases} \text{Yes} \\ \text{No} \end{cases}$

input

example of NP problem: 3SAT

CNF formula

3SAT: X is a YES instance if it describes a Boolean formula that is an AND of ORs where each clause has at most 3 literals and there is an assignment of variables that makes X true

$z_i, \neg z_i$

Instance:

$$X = (z_1 \underset{\text{OR}}{\vee} z_2 \underset{\text{OR}}{\vee} \neg z_3) \overset{\text{AND}}{\wedge} \boxed{(\neg z_1 \vee \neg z_3 \vee z_4)} \overset{\text{AND}}{\wedge} (z_2 \vee z_4) \wedge \cdots$$

$z_1, z_2, \ldots \Leftrightarrow$ variables

$\neg z_1, z_1, \neg z_2, z_2 \Leftrightarrow$ literals

Otherwise X is a NO instance

Is $3SAT \in NP$?

Questions to Ask Yourself to Prove $Q \in NP$

① What info would convince me that x is a Yes for Q?

x is YES for 3SAT:

$$(z_1 = T,\ z_2 = F,\ z_3 = F \ldots) \leftarrow y$$

② If given info in ①, how could I check if x is a yes for Q, quickly?

3SAT: Plug y in and test each clause

☆ You do not have to show how to find y ☆

☆ Only need to show how to verify y is solution to x ☆

$$x = z_1 \wedge \neg z_1$$

$$y = z_1 \leftarrow F/T$$

watch

Proof sketch that $3SAT \in NP$

[NP1]

Let $M(x, y)$

alg.
instance
potential solution

be the alg that

1. Check that x is a CNF formula with at most 3 literals per clause.
2. Check that y is a assignment of T, F to each variable.
3. Check that with assignment y, every clause of x is true.

And output 1 if all checks pass, 0 otherwise.

- If x is a yes instance, then a satisfying assignment y exists and $M(x,y)$ will pass all checks. Otherwise if x is a NO instance, at least one check will fail for any y.
- If 3-SAT is a Yes-instance with n variables, m clauses then $|x| = \Omega(m \log_2 n)$ so $M(x,y)$ can be run in $O(\text{poly}(|x|))$ time with brute force.

Formal(ish) Definition of NP

"Decision Problem"

A problem is in NP if

- YES-NO
- There is a polytime algorithm M s.t.
  - If x is a YES instance, $\exists y$ s.t. $M(x,y)=1$
  - If x is a NO instance, $\forall y$, $M(x,y)=0$

input x

$O(\text{poly}(|x|))$

"verifier"

"witness"

Group Work [NP1]

Hamiltonian Path Problem: X is a YES instance iff X describes adjacency matrix of a graph G with vertices s, t such that s.t. there is a path from s to t that goes through each vertex exactly once.

X =

	s	u	v	t
s	0	1	0	0
u	1	0	1	1
v	0	1	0	1
t	0	1	1	0

X =

	s	u	v	t
s	0	1	1	1
u	1	0	0	0
v	1	0	0	1
t	1	0	1	0

Show: $\Uparrow$ YES instance $\qquad$ $\Uparrow$ No instance

Prove: Hamiltonian Path $\in$ NP

- Describe $M(x, y)$
- Lower bound $|x|$ in terms of n (vertices), m (edges) $\rightarrow$ poly($|x|$) runtime for M
- Correctness ($\exists y$, $\forall y$)

i

- Names, Pronouns
- Dining Hall Hack

Group Work

Hamiltonian Path Problem: X is a YES instance iff X describes adjacency matrix of a graph G with vertices s, t s.t. there is a path from s to t that goes through each vertex exactly once.

X =

	s	u	v	t
s	0	1	0	0
u	1	0	1	1
v	0	1	0	1
t	0	1	1	0

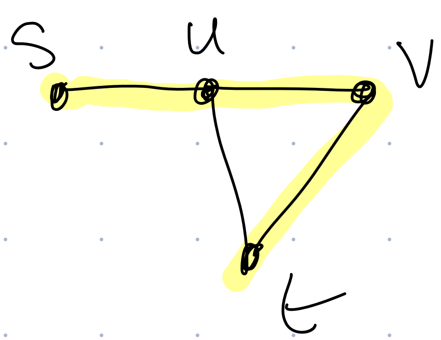
↑ YES instance

X =

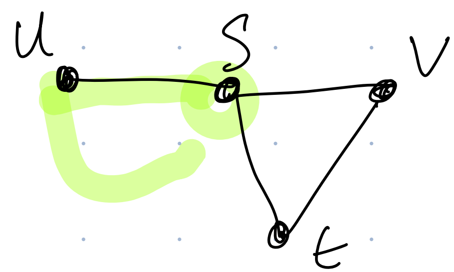
	s	u	v	t
s	0	1	1	1
u	1	0	0	0
v	1	0	0	1
t	1	0	1	0

↑ No instance

Show:



$y = (s, u, v, t)$



Ham Path $\in$ NP

- $M(x,y)$ check
  1. Check x is adj matrix with vertices s,t.
  2. y is a list of n vertices start at s, end at t.
  3. y has no repeats
  4. For each consecutive pair $(u,v) \in y$, (u,v) should be edge in G
- And output 1 if all checks pass, 0 otherwise.
  - If x is a Yes instance, then a path y exists and $M(x,y)$ will pass all checks. Otherwise if
  - X is a NO instance, at least one check will fail for any y.
  - If a yes instance has n vertices, m edges $|x| = \Omega(n^2)$ so $M(x,y)$ can be run in $O(\text{poly}(|x|))$ time with brute force