Grades so far

• An idea of eventual scale:
  - 2023 so far: median 79.9
  - 2022: mean 80.9; median 83.6
  - 2021: mean 85.7; median 86.9
  - 2020: mean 81.3; median 81.8
  - 2019: mean 82.0; median 84.0

Outline

• The class NP
  – alternate characterization of NP
• 3-SAT is NP-complete
• NP-complete problems: independent set, vertex cover, clique...

Back to 3SAT

• Remember 3SAT $\in$ EXP
  3SAT = \{formulas in CNF with 3 literals per clause for which there exists a satisfying truth assignment\}

• It seems hard. Can we show it is intractable?
  – formally, can we show 3SAT is EXP-complete?

Back to 3SAT

• can we show 3SAT is EXP-complete?
  • Don't know how to. Believed unlikely.
  • One reason: there is an important positive feature of 3SAT that doesn't seem to hold for problems in EXP (e.g. ATM):

  3SAT is decidable in polynomial time by a nondeterministic TM

Nondeterministic TMs

• Recall: nondeterministic TM
• informally, TM with several possible next configurations at each step
• formally, A NTM is a 7-tuple 
  \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\)
  where:
  – everything is the same as a TM except the transition function:
  \(\delta: Q \times \Gamma^* \rightarrow 2(Q \times Q \times \{L, R\})\)
Nondeterministic TMs

visualize computation of a NTM M as a tree

- nodes are configurations
- leaves are accept/reject configurations
- M accepts if and only if there exists an accept leaf
- M is a decider, so no paths go on forever
- running time is max. path length

The class NP

**Definition**: \( \text{TIME}(t(n)) = \{ L : \text{there exists a TM M that decides } L \text{ in time } O(t(n)) \} \)

\( P = \bigcup_{k \geq 1} \text{TIME}(n^k) \)

**Definition**: \( \text{NTIME}(t(n)) = \{ L : \text{there exists a NTM M that decides } L \text{ in time } O(t(n)) \} \)

\( \text{NP} = \bigcup_{k \geq 1} \text{NTIME}(n^k) \)

NP in relation to P and EXP

- \( P \subseteq \text{NP} \) (poly-time TM is a poly-time NTM)
- \( \text{NP} \subseteq \text{EXP} \)
  - configuration tree of \( n^k \)-time NTM has \( \leq b^n \) nodes
  - can traverse entire tree in \( O(b^n) \) time

NP is not known to be properly contained in P or EXP.

Poly-time verifiers

- \( \text{NP} = \{ L : \text{L decided by poly-time NTM} \} \)
- Very useful alternate definition:

**Theorem**: language \( L \) is in \( \text{NP} \) if it is expressible as:

\[ L = \{ x : \exists y, |y| \leq |x|^k, (x, y) \in R \} \]

where \( R \) is a language in \( P \).

- poly-time TM \( M \) deciding \( R \) is a "verifier"

Poly-time verifiers

- Example: 3SAT expressible as

\[ 3\text{SAT} = \{ \varphi : \varphi \text{ is a 3-CNF formula for which } 3 \text{ assignment } A \text{ for which } (\varphi, A) \in R \} \]

\( R = \{ (\varphi, A) : A \text{ is a sat. assign. for } \varphi \} \)

- satisfying assignment \( A \) is a "witness" of the satisfiability of \( \varphi \) (it "certifies" satisfiability of \( \varphi \))
- \( R \) is decidable in poly-time

Poly-time verifiers

\[ L = \{ x \mid \exists y, |y| \leq |x|^k, (x, y) \in R \} \]

**Proof**: \( (\Leftarrow) \) give poly-time NTM deciding \( L \)

phase 1: "guess" \( y \) with \( |x|^k \) nondeterministic steps

phase 2: decide if \( (x, y) \in R \)
Poly-time verifiers

\textbf{Proof:} (⇒) given \( L \in \text{NP} \), describe \( L \) as:
\[ L = \{ x \mid \exists y, |y| \leq |x|^k, (x, y) \in R \} \]

- \( L \) is decided by NTM \( M \) running in time \( n^k \)
- define the language \( R = \{ (x, y) : y \text{ is an accepting computation history of } M \text{ on input } x \} \)
- check: accepting history has length \( \leq |x|^k \)
- check: \( M \) accepts \( x \) iff \( \exists y, |y| \leq |x|^k, (x, y) \in R \)

Cook-Levin Theorem

- Gateway to proving lots of natural, important problems \( \text{NP} \)-complete is:

\textbf{Theorem (Cook, Levin):} \( 3\text{SAT} \) is \( \text{NP} \)-complete.

- Recall: \( 3\text{SAT} = \{ \phi : \phi \text{ is a CNF formula with 3 literals per clause for which there exists a satisfying truth assignment} \} \)

Boolean Circuits

- every function \( f: (0,1)^n \to (0,1) \) computable by a circuit of size at most \( O(n^2) \)
  - AND of \( n \) literals for each \( x \) such that \( f(x) = 1 \)
  - OR of up to \( 2^n \) such terms

\textbf{Theorem:} CIRCUIT-SAT is \( \text{NP} \)-complete

CIRCUIT-SAT = \( \{ C : C \text{ is a Boolean circuit for which there exists a satisfying truth assignment} \} \)

\textbf{Proof:}
- Part 1: need to show CIRCUIT-SAT \( \in \text{NP} \).
  - can express CIRCUIT-SAT as:
    \[ R = \{ (C, x) : C \text{ is a Boolean circuit and } C(x) = 1 \} \]
CIRCUIT-SAT is NP-complete

CIRCUIT-SAT = \{ C : C is a Boolean circuit for which there exists a satisfying truth assignment\}

Proof:
– Part 2: for each language A ∈ NP, need to give poly-time reduction from A to CIRCUIT-SAT
– for a given language A ∈ NP, we know A = \{ x | \exists y, |y| ≤ |x|^k, (x, y) ∈ R \} and there is a (deterministic) TM M_R that decides R in time g(n) ≤ n^c for some c.

Tableau (configurations written in an array) for machine M_R on input w = (x, y):

• height = time taken = |w|^c
• width = space used ≤ |w|^c

Important observation: contents of cell in tableau determined by 3 others above it:

| 0/? | 1 | 0
+-----+---+--
| 0/? | 0 | 0
| 0/? | 0 | 1
| 0/? | 1 | 0
| 1/? | 1 | 0

Can build Boolean circuit STEP
– input (binary encoding of) 3 cells
– output (binary encoding of) 1 cell
• each output bit is some function of inputs
• can build circuit for each
• size is independent of size of tableau

| 1 | 0 | 0
+---+---+--
| 0 | 1 | 0
| 0 | 0 | 1

Tableau for M_R on input w = (x, y):

• |w|^c copies of STEP compute row i from i-1

This circuit C_{M_R} has inputs w_1w_2...w_n and C(w) = 1 iff M_R accepts input w.

Size = O(|w|^c)
CIRCUIT-SAT is NP-complete

- recall: we are reducing language A:
  \[ A = \{ x \mid \exists y, |y| \leq |x|, (x, y) \in R \} \]
to CIRCUIT-SAT.
- \( f(x) \) produces the following circuit:
  - hardwire \( x \)
  - leave \( y \) as variables

\[ 1 \text{ iff } (x,y) \in R \]

Circuit \( C_{\text{SAT}} \), \( w \)

CIRCUIT-SAT is NP-complete

- is \( f(x) \) poly-time computable?
  - hardcode \( M_R, k \) and \( c \)
  - circuit has size \( O(|w|^2); |w| = |(x,y)| \leq n + n^2 \)
  - each component easy to describe efficiently from description of \( M_R \)

- YES maps to YES?
  - \( x \in A \Rightarrow \exists y, M_R \text{ accepts } (x, y) = f(x) \in \text{CIRCUIT-SAT} \)
- NO maps to NO?
  - \( x \notin A \Rightarrow \forall y, M_R \text{ rejects } (x, y) = f(x) \notin \text{CIRCUIT-SAT} \)