Today

- Introduction
- Administrivia
- What is this course about
- Model of Computation

CSS.203.1
Computational Complexity
- Lecture #1
Instructor: Prahladh Harsha

Simplicity does not precede complexity, but follows it
— Alan Perlis

The cube, is at the same time, a symbol of simplicity and complexity
— Erno Rubik

—

Grading Policy:

- 6 psets (50%) (best 5 of 6)
- 1 mid term (20%)
- 1 final exam (20-25%)
- in class presentations / quizzes — 5% - 10%

Book: Arora - Barak · Computational Complexity

— Lecture Notes. Sudan, Trevisan.

—

What's this course about?

Theory of Computation

- Computable / Not computable.

"Easy / Hards" - computation.

→ What does it mean to be an "easy / hard" problems?

→ Different types of resources (time, space, randomness, communication / interaction)

→ Interplay across the resources

→ Compare & contrasting interesting problems

→ Interesting Problems (Phenomena / classes)

Interesting Problems

1. Parity : Given $x \in \{0,1\}^n$, compute $\bigoplus_{i=1}^{n} x_i$.
2. Integer Multiplication

Given a, b in binary notation of n bits each.

Compute $a \cdot b$.

→ $O(n^2)$

$O(n \cdot \log n \cdot \log\log n)$

3. **Connectivity**: Given a graph $G = (V, E)$ and 2 s - source, t - target

Is there a path $s \rightsquigarrow t$?

(directed / undirected)

4. **Matching**: Given a bipartite graph $G = (L, R, E)$, does there exist a perfect matching? If so, find one.

5. **SAT**: (satisfiability): Given a Boolean formula φ is it satisfiable?

6. **#SAT**: Given a Boolean formula φ, find the number of satisfying assignments?

7. **Permanent.** · Given a matrix $M \in \{0,1\}^{n \times n}$

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & \cdots & & \\ & & & \\ m_{n1} & & & \end{bmatrix}$$

$$\text{perm}(M) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} m_{i,\sigma(i)}$$

(unsigned determinant)

8. **CNF minimization.** Given a Boolean formula φ (in CNF) & m

Qn: Does there exist an "equivalent" CNF ψ of (i) $\psi \equiv \varphi$

(ii) $\text{size}(\psi) \leq m$

(Equivalent: φ & ψ are equivalent if $\forall a \in \{0,1\}^n$, $\varphi(a) = \psi(a)$)

If there exists an alg for CNF-minimization, then can solve SAT.

Problem 8 is at least as hard as Problem 5

[Cook-Levin] SAT is NP-complete

[Valiant] #SAT & Permanent are "equivalently" hard

[Stockmeyer + Toda] SAT $\leq$ CNF-minimization $\leq$ #SAT

What we would like to show

Problem A is <u>hard</u>

Unfortunately, we are far from doing this.

Relate hardness of one problem to the hardness of another problem.

<u>Reduction</u>:

A B

A "is no harder than" B

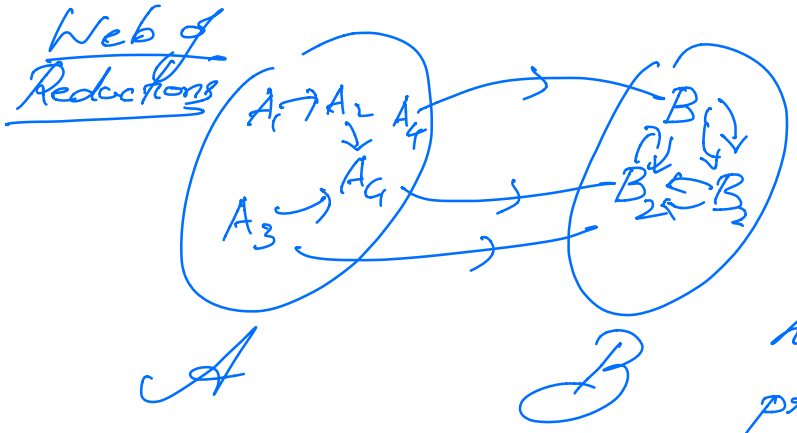
<u>Sub-routine Reductions</u>: (Turing Reductions)

Can use a sub-routine for B to solve A

<u>Many-to-one Reductions</u> (Karp Reductions)

A very specialized kind of sub-routine redn

Web of Reductions



Problems in B are equivalently hard & capture the hardness of all problems in $A \cup B$.

B - complete problem. $A \cup B$.

Model of Computation:

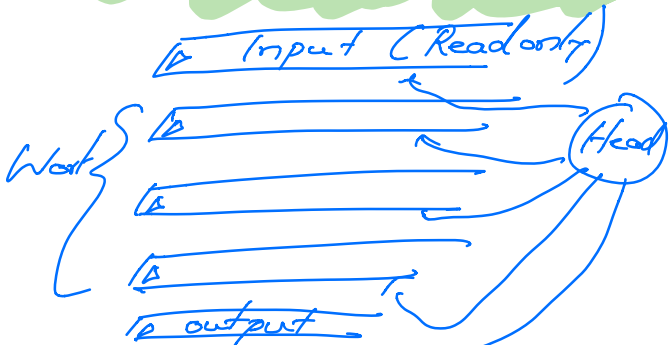
Hardness / Easiness - model independent

Model - Turing Machine

Captures all reasonable models of computation!

Turing Machine (TM)

(k, Γ, Q, δ)



k - # tapes

Γ: alphabet $= \{0, 1, \triangleright_{start}, \square_{blank}\}$

Q: state space $= \{q_{start}, q_{halt}\}$

δ: Rules of TM.

$\delta: Q \times \Gamma^k \longrightarrow Q \times \Gamma^{k-1} \times \{L, R, S\}^k$

TM: $M = (k, \Gamma, Q, \delta)$.

Input – Read only (L, R, S)

Work + output tapes – RW (L, R, S)

M "compute" $f: \{0,1\}^* \to \{0,1\}^*$

If the machine M is initialized $\forall x \in \{0,1\}^*$

w/ $\triangleright x \square\square$ → input tape

+ blanks on the other tapes w

+ q_s – start state.

↓ Applies transition fn δ repeatedly

Eventually does go into q_h – halt state

+ if that has $f(x)$ written on the output tape

M computes $f: \{0,1\}^* \to \{0,1\}^*$ in time T

$T: \mathbb{N} \to \mathbb{N}$

On inputs of length n, the m/c takes $T(n)$ steps.

(the transition fn δ is

applied almost $T(|x|)$ on ~~every~~ each i/p x.

What are $T: \mathbb{N} \to \mathbb{N}$ allowed?

eg: $T(n) = n^2$, 2^n, $n \log n$

$T(n)$ = number of steps taken by the n^{th} TM on n^{th} input.
↳ perverse

T- Time constructible

$T: \mathbb{N} \to \mathbb{N}$ is time constructible.

(i) $T(n) \geq n$

(ii) $\exists$ a TM M that computes the fn $x \mapsto \underbrace{T(|x|)}_{\text{in binary}}$ within $T(|x|)$ time steps.

Specific defn of TM:

- Amazingly Robust
- Alphabet (finite)

Claim: Suppose M computes f in time T w/ alphabet Γ.

then $\exists$ a M' that computes f in time $C \log(T)$ w/ Boolean alphabet

tapes - does not matter (as long as $k \geq 2$)

Claim: Can simulate a k-tape TM using a single TM w/ at most a quadratic.

Single tape
↳ only input tape
R/W tape

TM - specific def's
- alphabet
- # tapes
- single-sided / double-sided

} robust

Next time:
- Universal TM.
- Time Complexity Classes (P, NP & polytime reducs)