Administration:

- HW: is easy this week—mainly probability background
- Optional problems: no credit

Complexity note

- model assumes source of random bits
- we will assume primitives: biased coins, uniform sampling
- in homework, see equivalent

Adelman's Theorem.

Consider RP (one sided error)

- Does randomness help?
  - In practice YES
  - in one theory model, no
  - in another, yes!
  - in another, maybe

  - Size n problems (2^n of them)
  - matrix of advice rows by input columns
  - some advice row **witnesses** half the problems.
  - delete row and all its problems
  - remaining matrix still RP (all remaining rows didn't have witness)
  - halves number of inputs. repeat n times.

  Result: on RP of size n, exists n witnesses that cover all problems.

  - polytime algorithm: try n witnesses.
  - Nonuniformity: witnesses not known.
  - $RP \subseteq P/poly$

oblivious versus nonoblivious adversary and algorithms.

Yao's Minimax Principle

How do we know our randomized algorithm is best possible?
Review tree evaluation.
Lower Bound
Game Theory

- Zero sum games. Scissors Paper Stone. Roberta, Charles.
- Payoff Matrix M. Entries are (large) strategies. chess.

Optimal strategies

- row wants to maximize, column to minimze
- suppose Roberta picks i. Guarantees $\min_j M_{ij}$.

- (Pessimistic) R-optimal strategy: choose i to $\max_i \min_j M_{ij}$.
- (Pessimistic) C-optimal strategy: choose j to $\min_j \max_i M_{ij}$.

When C-optimal and R optimal strategies match, gives **solution** of game.

- if solution exists, knowing opponents strategy useless.
- Sometimes, no solution using these **pure** strategies

Randomization:

- **mixed** strategy: distribution over pure ones
- R uses dist p, C uses dist q, expected payoff $p^T M q$
- Von Neumann:
$$\max_p \min_q p^T M q = \min_q \max_p p^T M q$$
that is, always exists solution in mixed strategies.
- Once p fixed, exists optimal pure q, and vice versa

Yao's minimax method:

- Column strategies algorithms, row strategies inputs
- payoff is expected running time
- randomized algorithm is mixed strategy
- optimum algorithm is optimum randomized strategy
- worst case input is corresponding optimum pure strategy
- Thus:
  - worst case expected runtime of optimum rand. algorithm
  - is payoff of game
  - instead, consider randomized inputs
  - payoff of game via optimum pure strategy
  - which is detemrinistic algorithm!
- Worst case expected runtime of randomized algorithm for any input equals best case running time of a deterministic algorithm for worst distribution of inputs.
- Thus, for lower bound on runtime, show an input distribution with no good deterministic algorithm

Game tree distribution.

- input distribution: each node 1 with probability $p = \frac{1}{2}(3 - \sqrt{5})$.
- every node is 1 with probability p
- lemma: any deterministic alg showld finish evaluating one child of a node before doing other: *depth first pruning algorithm*
- Such algorithm has probability p of finding 1 on first child, so
$$W(h) = W(h-1) + (1-p)W(h-1) = (2-p)^h = n^{0.694}$$

Game tree evaluation lower bound.

- Recall Yao's minimax principle.
- lemma: any deterministic alg showld finish evaluating one child of a node before doing other: *depth first pruning algorithm.* proof by induction.
- input distribution: each leaf 1 with probability $p = \frac{1}{2}(3 - \sqrt{5})$.
- every node is 1 with probability p
- let $T(h)$ be expected number of leaves evaluated from height h.
- with probablity p, eval one child. else eval 2.
- So
$$T(h) = pT(h-1) + 2(1-p)T(h-1) = (2-p)^h = n^{0.694}$$