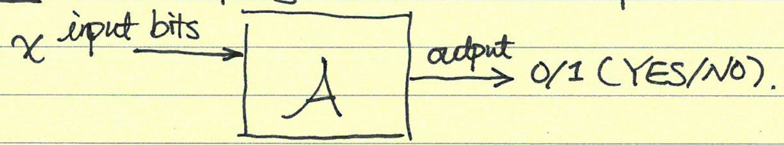
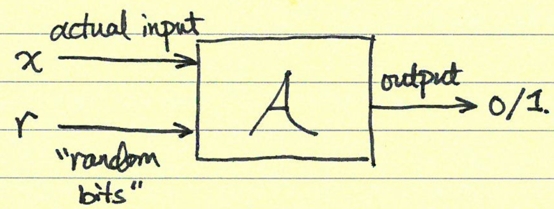
Class of Randomized Algorithms and Derandomization.

Deterministic algorithms. (For simplicity, we focus on decision problems now)



(ideally we want A to run in polynomial time, say time $\le n^k$)

Randomized algorithms. A is still deterministic, but has "random bits"



Example. Miller-Rabin Primality testing: x is the # to be tested.

A solves a problem in "BPP" if $\forall x$

$$\Pr_r[A(x,r) \text{ correct}] \ge 3/4$$

Remark 1 "$3/4$" can be any $c \in (1/2, 1)$. To make it $1-\varepsilon$, we run $A(x,r)$ $O(\log \frac{1}{\varepsilon})$ times indep., and take the majority vote.

Remark 2 If A only requires $O(\log n)$ random bits, it's trivial to make A deterministic. — Simply try all $2^{O(\log n)} = \text{poly}(n)$ possible r's, and take the majority vote.

Derandomization. (Is BPP = P?) How to make A deterministic even if it uses $\omega(\log n)$ random bits?

Pseudorandom Generator. (PRG) Let $\mathcal{C}$ be a class of fcn's $f: \{0,1\}^n \to \{0,1\}$. $G: \{0,1\}^\ell \to \{0,1\}^n$ $(\ell < n)$ is an ε-PRG for $\mathcal{C}$ ~~if~~ with seed length ℓ if.

$$\forall f \in \mathcal{C}: \left| \Pr_{s \sim \{0,1\}^\ell}[f(G(s)) = 1] - \Pr_{r \sim \{0,1\}^n}[f(r) = 1] \right| < \varepsilon. : \text{"}G\ \varepsilon\text{-fools } \mathcal{C}\text{"}$$

Typically, want $G(s)$ computable in poly(n) time (deterministically).

Intuition $\mathcal{C}$ not able to distinguish between distrib. $\{G(s)\}_{s \sim \{0,1\}^\ell}$ and uniform distrib. $\{0,1\}^n$. However $\{G(s)\}$ has a much smaller support.

Example. Say A runs in n^{10} time, uses n random bits. Let $\mathcal{C}$ be $\{f: \{0,1\}^n \to \{0,1\},\ f$ computable in n^{10} time$\}$. If G $.1$-fools $\mathcal{C}$, then

$$\Pr_{s \sim \{0,1\}^\ell}[A(G(s)) \text{ correct}] \ge \Pr_{r \sim \{0,1\}^n}[A(r) \text{ correct}] - .1 \ge 3/4 - .1 = .65$$

A deterministic alg. to enumerate s and take maj. vote runs in $2^\ell \text{poly}(n)$ time.

If $l = O(\log n)$, the algorithm solves A in P.

<u>Theorem [Impagliazzo-Widgerson'97]</u> Suppose $\forall m\ \exists h_m : \{0,1\}^m \to \{0,1\}$ computable in time 2^{100m}, but not in time $2^{.001m}$, then there is a PRG ε-fools all poly-time algorithms with seed length $O(\log n)$, i.e. BPP = P. (The assumption is stronger than P $\neq$ NP, but believable).

<u>Intuition.</u> A function hard to compute $\Rightarrow$ looks random to Turing Machines with less time resource $\Rightarrow$ fools these TMs.

<u>k-wise Independent PRGs.</u> $G : \{0,1\}^l \to \{0,1\}^n$ is k-wise indep. if

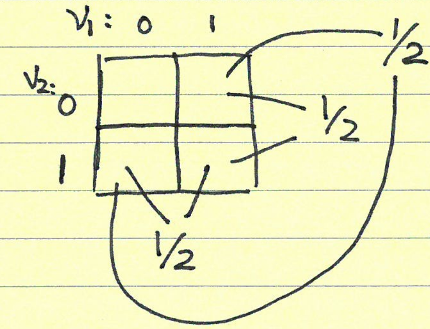
* $\forall i \in [n]$ $\Pr_s[(G(s))_i = 1] = 1/2$
* $\forall 1 \le i_1 < i_2 < \cdots < i_k \le n$ the distrib. $\{(G(s))_{i_1}, (G(s))_{i_2}, \ldots, (G(s))_{i_k}\}_s$ is uniform on $\{0,1\}^k$

<u>Constructing pairwise indep. PRGs.</u> $G : \{0,1\}^l \to \{0,1\}^{2^l - 1}$ defined as

$$[G(s)]_v = \langle s, v\rangle \bmod 2 \quad \text{for all } v \in \{0,1\}^l,\ v \neq \vec{0}$$

<u>Proof.</u> $\forall v \neq \vec{0}$: $\Pr_s[\langle s,v\rangle \bmod 2 = 1] = 1/2$

$$\forall v_1 \neq v_2 : \Pr_s[\langle s,v_1\rangle \bmod 2 \neq \langle s,v_2\rangle \bmod 2]$$
$$= \Pr_s[\langle s, v_1+v_2\rangle \bmod 2 \neq 0] = 1/2$$



<u>Recall</u> Hadamard Code.

<u>Theorem</u> [Alon-Babai-Itai '85] $\forall k \le n$, ~~prime power q~~, $\exists$ poly(n)-time computable k-wise indep. generator with $l = \lfloor \frac{k}{2} \rfloor \log n + O(1)$.

<u>Application</u> Derandomize the following algorithm for Max-Cut.

<u>MaxCut.</u> Given $G = (V,E)$, find $S \subseteq V$ to maximize $|\text{edges}(S, V-S)|$

<u>Alg.</u> For each $i \in V$, toss $r_i \in \{0,1\}$, $i \in S$ iff. $r_i = 1$

<u>Analysis.</u> $\mathbb{E}\,|\text{edges}(S,V-S)| = \mathbb{E}_r \sum_{(i,j)\in E} 1[r_i \neq r_j]$

$$= \sum_{(i,j)\in E} \Pr_r[r_i \neq r_j] = \sum_{(i,j)\in E} \frac{1}{2} = \frac{|E|}{2} \leftarrow \text{cut at least 50\% edges not bad.}$$

$\uparrow$ linearity of expectation $\qquad \uparrow$ pairwise indep.

<u>Observation.</u> $r \in \{0,1\}^n$ be pairwise indep. suffices for the analysis.
use $r \leftarrow G(s)$ where $s \in \{0,1\}^{\log n}$, G pairwise indep.
enumerate s in polynomial-time.

<u>ε-biased Generators.</u> $G: \mathbb{F}_2^{\ell} \to \mathbb{F}_2^n$ is an ε-biased generator if

$$\forall w \in \mathbb{F}_2^n,\ w \neq 0, \quad \Pr_{s \sim \mathbb{F}_2^{\ell}}\left[w \cdot G(s) = 1\right] \in \left[\tfrac{1}{2} - \tfrac{\varepsilon}{2}, \tfrac{1}{2} + \tfrac{\varepsilon}{2}\right]$$

— it $\frac{\varepsilon}{2}$-fools all deg-1 / linear functions.

<u>Theorem</u> [NN'93] $\ell = O(\log \frac{n}{\varepsilon})$ achievable w/ G poly-time computable.
[AGHP'92] $\ell = 2\log\frac{n}{\varepsilon} + O(1)$, $O(n^2/\varepsilon^2)$-time computable

<u>Application.</u> Input: $A, B, C \in \mathbb{F}_2^{n \times n}$

Goal: Check $AB \stackrel{?}{=} C$ in $O(n^2)$ [input size] time.

Alg: Choose $y \sim \mathbb{F}_2^n$ uniformly, check if

$$\underbrace{(AB)y}_{\substack{\| \\ \underbrace{A\underbrace{(By)}_{O(n^2)\text{ time}}}_{O(n^2)\text{ time}}}} = \underbrace{Cy}_{O(n^2)\text{ time}}$$

Analysis: when $AB = C \Rightarrow \Pr[(AB)y = Cy] = 1$

when $AB \neq C \Rightarrow D = AB - C$ has ≥ 1 non-zero row, namely d

$$\Pr[(AB)y = Cy] = \Pr[Dy = 0] \leq \Pr[d \cdot y = 0] = 1/2$$

[Can repeat w/ several y to gain high confidence]

Uses $O(n^2)$ time. n random bits.

If y is output of a $.1$-biased gen. $\Pr[d \cdot y = 0] \leq 1/2 + \frac{.1}{2} = .55$

$\rightarrow$ $O(n^2)$ time, $O(\log n)$ random bits. (using [AGHP'92])