Lecture 24:

Hardness vs. Randomness
A lemma for next time:

\[ \text{def } I = \{ I_1, \ldots, I_m \} \subseteq [l] \text{ is } (l, n, d) \text{-design } \quad (d > n > d) \]

\[
\begin{align*}
\text{if } & |I_j| = n \quad \forall j \\
\text{or } & |I_j \cap I_k| \leq d \quad \forall j \neq k
\end{align*}
\]

\[ \text{Thm. } \text{An algorithm running in } 2^{o(n)} \text{ time s.t. for } n > d, \quad l > 20n^2/d \]

which outputs \((l, n, d)\)-design

\[
\text{st. } m = 2^{d/10}
\]

\[ \text{Pf. } \]

Greedy - best parameters, use prob method to show can progress:

**GreedyAlg:** after having \( I_1, \ldots, I_l \) for \( l \leq 2^{d/10} \),

search all subsets to find \( I^* \) s.t. \( |I^* \cap I_j| \leq d \quad \forall j \in [l] \)

**runtime:** \( \text{poly}(m) \cdot 2^l \)

**Why doesn't it get stuck?**

if pick \( I^* \) randomly:

\[ \text{prob } x \in [l] \text{ gets chosen } = \frac{2^n}{l} \]

(and truncate later)

\[
\begin{align*}
E[I^*|I_j] &= 2m \\
Pr \left[ |I^*| \geq n \right] &\geq 0.9 \\
E[I^*|I_j] &= m^2 \\
Pr \left[ |I^* \cap I_j| = d \right] &\leq \frac{1}{2} \
\end{align*}
\]

Since \( m \leq 2^{d/10} \), via union bound, with prob \( \geq 0.4 \)

\( I^* \) will be good.
Other constructions:

based on polynomials -

computable in parallel, small space, low sequential time

Recall our goal: Derandomizing algorithms

\[
\text{Input } x \rightarrow \text{Program} \rightarrow \text{Output}
\]

\[
\text{total time } = \frac{\# \text{seeds}}{2^a} \times \text{runtime of PRG} + \text{runtime of program}
\]

- to derandomize allptime algs, need PRG
  which takes \(O(\log n)\) bits, outputs \(nc\) bits which \(\leq U_{nc}\)
  \& runs in \(\text{poly}(n)\) time

- today: derandomize parallel algorithms,
  i.e., need PRG which outputs bits that look uniform to parallel algs

- more generally: hard on average fcnrs (on \(\leq S(n)\) size, get advantage \(\frac{1}{2^k}\))
  \(\Rightarrow\) PRGs (bits of size \(\leq S(\log n, \kappa)\) get adv \(\leq \frac{1}{10}\))
Def. $f : \{0,1\}^l \rightarrow \{0,1\}$ is $(\varepsilon, \varepsilon)$-average case hard if $A$ nonuniform in time $t^\varepsilon(l)$

$$\Pr_{x \in \{0,1\}^l} [A(x) = f(x)] \quad \text{for large enough } l$$

$$\leq \frac{1}{2} + \varepsilon(l) \quad \text{pick } \varepsilon(l) < \frac{1}{2k(l)}$$

so

$$\leq \frac{1}{2} + \frac{1}{k(l)}$$

$f$ is $\varepsilon$-average case hard if for nonuniform $A$ in time $t^\varepsilon(l)$, adv $A(1^n) \approx t^\varepsilon(l)$

Thm.

If $f : \{0,1\}^l \rightarrow \{0,1\}$ is $(\varepsilon, \varepsilon)$-average case hard

Then $G(y) = y \circ f(y)$ is $(\varepsilon, \varepsilon)$-PRG

up

passing through

pf. as for HC$\beta$

\[ \text{How to stretch?} \]

Define $N \cdot N$ generator...
**Def. Nisan-Wigderson generator**

Given $(y, n, d)$- design $T = \{T_1, \ldots, T_m\} \subseteq [d]$,

$G : \Sigma^* \rightarrow \Sigma_1 \Sigma^m$

is $G(x) = f(x|T_1) \circ f(x|T_2) \circ \cdots \circ f(x|T_m)$

where $f_i(x) = f(x|T_i)$

**Thm [NW]** (1) $\exists f : \Sigma^* \rightarrow \Sigma_1 \Sigma^m$, st. $f \in E = \text{DTIME} \left(2^n \right)$

st. $f$ is $\varepsilon$- average case hard

(2) $\exists (y, n, d)$ design with $m$ sets, constructable in time $2^{o(n)}$

st. $m = 2^{\Omega(n)}$, $l = \Omega(n^2/d)$, $n > d$

then $G$ is $\frac{c}{m}$-PRG against nonuniform time $m$. 

Can think of $\varepsilon = \frac{1}{10}$
pf.

if $G$ not $\frac{1}{m}$-PRG against time $m$,

exists n.b. predictor $P$ s.t.

$$\Pr_{x \sim \{0,1\}^n} \left[ \prod_{i \leq m} P(f_i(x) \cdots f_{X}^{x} = f_{X}^{x} | x) \right] = \frac{1}{2} + \frac{\varepsilon}{m}$$

time $(P) = time(T) + O(m)$

Plan: use this to approx $f$ with $\frac{\varepsilon}{m}$ adv in $O(\ell(n))$ time
to contradict $f$'s hardness where $m \ll \ell(n)$

As usual, averaging $\Rightarrow$ exists st. attain expectation

$\Rightarrow$ exists choice of bits of $x$ not in $I_{x^*}$ attaining expectation

call it $z$

notion $Y \leftarrow x$ with bits in $I_{x^*}$ set to this choice $z$ & others picked randomly

so

$$\Pr_{y} \left[ \prod_{i \leq m} P(f_i(y) \cdots f_{X}^{x} = f_{X}^{x} | y) \right] = \frac{1}{2} + \frac{\varepsilon}{m}$$

properties of

$(\delta, \eta, d)$-design

give this

since $|I_{x^*} \cap I_{y}| \leq d$

bits of $y$

each depends on $\leq d$

since depend on $\leq d$ bits,
and $f \in E$, can compute
each $f_j$ in time

$2^d$ or with ckt

that has encoded lookup table

not when trying to prove $P=\mathsf{BPP}$
$A(y) = P(f_1(y), f_2(y), \ldots, f_{x+1}(y))$

- predicts $f_{x+1}(y)$ with $\text{adv} \geq \frac{\epsilon}{m} = \frac{1}{10} \cdot \frac{d}{10}$

**Runtime**

- $\tilde{O}(d^{2d}) \cdot O(m) + \tilde{O}(m)$ to find "design" bits
- $O(m)$ such bits
- $\frac{d}{10}$
- $2$

Set $d = \log \frac{t}{\epsilon}$

So it is $\tilde{O}(t^{\frac{1}{10}})$

but $\tilde{O}(d^{2d}) \cdot O(m) + O(m) = \tilde{O}(t^{\frac{1}{10}}) \cdot O(t^{\frac{1}{10}}) + \frac{t}{2}$

$\leq t$

Which contradicts hardness of $f$