## Today

- Distributional Problems.
- Hardness of the Permanent on Random instances.
- Average-case hardness within NP.



## Distributional Problems

- Problems come in two parts $(\Pi, D)$.
  - $\Pi$ is a problem to be solved.
  - $D$ is a distribution on instances.
- Avg=P: Goal is to solve all but $\delta$ fraction of the problems in time polynomial in input length and $\frac{1}{\delta}$.
- DNP: Problem $\Pi$ described by relation $R$ and goal is to find, given $x$, a $y$ such that $R(x, y)$.



## Today

- If the permanent, on the "uniform" distribution, is in Avg-P, then $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{BPP}$.
- A complete problem (under what reductions?) for DNP.



## The permanent

- Let $S_n = \{$ bijections $\pi : [n] \to [n]\}$.
- For matrix $M$,

$$\mathrm{perm}(M) = \sum_{\pi \in S_n} \prod_{i=1}^{n} M_{i\pi(i)}.$$

- Valiant's Theorem: Permanent is complete for #-P.

## Uniform Distribution

- Given $n$, pick $p$ at random from $\{1, ..., n^3\}$.
- Pick entries of $n \times n$ matrix $M$ independently, uniformly at random from $\mathbb{Z}_p$.



## Lipton's Theorem

- If Permanent on Uniform distribution is in Avg-P, then $\mathsf{P}^{\#\mathsf{P}}$ is in BPP.
- Outline: Will show how to compute the permanent of any 0/1 matrix in BPP, if $P$ computes permanent on uniform distribution w.p. more than say $1 - n^{-10}$ in poly time.
- Given 0/1 matrix $A$ whose permanent we wish to compute. Will generate $(p_1, M_1), (p_2, M_2), \ldots, (p_m, M_m)$ such that $(p_i, M_i)$ is distributed uniformly (but not independently!) and solving the problem on all $m$ instances will give the answer for $A$.



## Step 1: Reduce to random $p$

- Pick $p_1, \ldots, p_k$ at random from $[n^3]$ till $\mathrm{lcm}(p_1, \ldots, p_k) > n!$.
- Note computing $\mathrm{perm}(A) \bmod p_i$ for all $i$ gives $\mathrm{perm}(A)$ modulo their LCM which is the permanent of A.
- So suffices to compute $\mathrm{perm}(A) \bmod p_i$.



## Computing modular permanent

- Wish to compute $\mathrm{perm}(A) \bmod p$.
- Pick $R \in \mathbb{Z}_p^{n \times n}$ at random.
- Let $M_x = A + xR \pmod p$. Note $M_x$ is random for $x \neq 0$.
- Compute permanent $\bmod p$ of $M_1, \ldots, M_{n+1}$. Since $B$ computes each value w.p. $1 - \delta$, it computes the whole sequence w.p. at least $1 - (n+1)\delta$.
- Claim: Have enough info to compute permanent of $A$!

## Computing modular permanent (contd.)

- Claim: Have enough info to compute permanent of $A$!
- Proof: Consider $\mathrm{perm}(M_x)$, where $x$ is a variable.
- This quantity is a polynomial in $x$ of degree $n$.
- $\mathrm{perm}(A)$ is the constant term of this polynomial.
- Can interpolate for the polynomial from its value at $n+1$ places.
- Conclude: If $B$ computes $(p_i, M_i)$ correctly



w.p. $1-\delta$, then we get answer correctly w.p. at least $1-n^4\delta$ on worst-case.



## Reductions between DNP problems

- Know what it means to reduce $\Pi_1$ to $\Pi_2$. But how to reduce $(\Pi_1, D_1)$ to $(\Pi_2, D_2)$?
- Consider even the simple case: Reduce $(\Pi, D_1)$ to $(\Pi, D_2)$. When is this trivial?
  - For $\alpha \geq 1$, $D_2$ $\alpha$-dominates $D_1$ if for every $x$ $\Pr_{D_2}[x] \geq \Pr_{D_1}[x]/\alpha$.
  - $A$ is $\delta$-good for $(R, D_2)$ implies $A$ is $(\alpha \cdot \delta)$-good for $(R, D_1)$.
  - $D_1$ dominates $D_2$ and $(\Pi, D_2)$ in Avg-P, implies $(\Pi, D_1)$ in Avg-P.
- $(\Pi_1, D_1)$ reduces to $(\Pi_2, D_2)$ if there exists a polynomial time function $f$ such that solving $\Pi_2$ on $f(x)$ solves $\Pi_1$ on $x$, and $D_2$



dominates the distribution of instances of $\Pi_2$ as induced by $f$ on $D_1$.

## Impagliazzo-Levin Lemma

Lemma: There exists a (essentially uniform) distribution $U$ such that for every $R, D$ there exists a relation $R'$ such that $(R, D)$ reduces to $(R', U)$.

Basic idea:

- Will try to make the relation $R'$ be $R$ composed with $D$.
- Need to specify $z$ in domain of $R$ given $x = D(z)$.
- Can't ask to invert $D$ — may be hard.
- So specify $z$ essentially by $x$ and an index $w \in \{0,1\}^k$ assuming $D^{-1}(x)$ has about $2^k$ members.
- $(x, w)$ does specify such a $z$, provided we pick $x$ according to $D(z)$. But don't get uniformity!
- So hash $(x, w)$ down to $n$-bit string $u$.





## Details

- Instances of $R'$ are tuples $(u, k, h_1, h_2)$ where $u \in \{0,1\}^n$, and $h_1 : \{0,1\}^n \to \{0,1\}^k$ and $h_2 : \{0,1\}^{n+k} \to \{0,1\}^n$ are nice pairwise independent has functions.
- $R'((u, k, h_1, h_2), (z, y))$ if $u = h_2(D(z), h_1(z))$ and $R(D(z), y)$.
- Distribution on instances $D_1 = U$ is the following: $u \in_U \{0,1\}^n$, $k \in_U \{0, \ldots, n\}$, $h_1, h_2$ are uniform from their families.



## Reduction

Need to reduce $(R, D)$ to $(R', U)$.

Reduction: Given $x$, guess $k$, and string $w$ (supposedly $k = \log |D^{-1}(x)|$ and $w = h_1(z)$ where $z \in_U D^{-1}(x)$). Pick $h_1$, $h_2$ uniformly, and then let $u = h_2(x, w)$. Output $(u, k, h_1, h_2)$.

Claim: Instances being generated essentially according to $U$.

Main step in proof: If we guess $k$ correctly, then very likely there is a unique $z$ such that $D(z) = x$ and $h_1(z) = w$.

## Soundness of reduction

- Related distribution $D_2$ on $R'$: Pick $z \in_U \{0,1\}^n$ and let $k = \log_2 |D^{-1}(D(z))|$. Let $h_1, h_2$ be uniform on their domain and let $u = h_2(D(z), h_1(z))$.
- Claim 1: $(R', D_2)$ is at least as hard as $(R, D)$.
- Claim 2: $D_1$ $n$-dominates $D_2$.
- Details left to the reader.



## A DNP-complete problem

- Easy to massage above into a relation $R''$ and distribution $U'$ which is actually uniform on its domain.
- But still don't have a single hard problem (i.e., relation and distribution).
- Use the universal relation [Levin].
- Hard problem: $R_U$ has as instances pairs $(R, x)$. $R_u((R, x), y)$ holds if $R(x, y)$.
- Claim $R_U$ on uniform distribution on inputs is at least as hard as $(R'', U)$ - since with probability $1/2^{|R''|}$ (a constant) we will generate $R''$ as the relation to be solved.



## Interlude

- Now have a theory of average-case hardness for problems in NP.
- How does it relate to worst-case hardness?
- Wide open.
- Known techniques relating the two don't seem to work [Feigenbaum-Fortnow]. (Does not rule out all reductions - only known forms.)
- Can we say anything within NP?

Ajtai'96 : Shows that worst-case instances of an "empirically" hard problem can be reduced to random instances of a related problem.



- Major breakthrough!

## Lattice problems

- Defn: Lattice $L$ in $\mathbb{R}^n$ is a discrete additive subset of $R^n$.
  - Discrete: Exists $d > 0$ such that for every point $x \in L$, the ball of radius $d$ around $x$ contains only one point ($x$) from $L$.
  - Additive: $x, y \in L$ implies $x + y, x - y \in L$.



## Specifying a lattice

- Primal specification: By basis: $b_1, \ldots, b_m \in \mathbb{R}^n$ (for $m \leq n$), $b_i$'s linearly independent, and lattice $L = \{\sum_{i=1}^{m} z_i b_i | (z_1, \ldots, z_m) \in \mathbb{Z}^m\}$.
- Dual specification: By vectors: $b_1^*, \ldots, b_m^* \in \mathbb{R}^n$ (for $m \geq n$), and lattice $L = \{\mathbf{v} \in \mathbb{R}^n | \forall j, \langle \mathbf{v}, \mathbf{b}_j^* \rangle \in \mathbb{Z}\}$.
- Can go from one rep'n to another algorithmically.



## Lattice problems

- Given lattice $L$, compute shortest non-zero vector in in lattice. Was open for long time, till [Ajtai] showed it to be NP-complete for randomized reductions.
- Given lattice $L$ and target vector $t \in \mathbb{R}^n$ compute nearest lattice point to $t$. (Long know to be NP-hard.)
- Given lattice, find short basis.
- Best algorithmic result: Can find $2^{o(n)}$ approximation for all above problems in poly time, for $n$-dimensional lattice.
- Shortest vector problem/Closest vector problem are of fundamental interest:



  - Used in factoring polynomials over integers [LLL].
  - Important case of Diophantine approximations.
  - Used widely in cryptanalysis.
  - Now becoming a basis for cryptography [Ajtai-Dwork].

## Ajtai's theorem

Roughly, gives approximation problems $R, R'$ and distribution $D$ such that an avg-P solution to $(R, D)$ implies a RP algorithm for $R'$.

- $R'$: Instance is a pair — a lattice $L$ and a bound $M$ with the promise that there exists a basis for $L$ with vectors of length at most $M$. Witness is a basis $b_1, \ldots, b_m$ where all vectors have length at most $\mathrm{poly}(n) \cdot M$.
- $R$: Instance is a pair — lattice $L$ given by dual vectors $b_1^*, \ldots, b_m^*$ and a bound $N$ with the promise that $L$ has a vector of length at most $N$. Witness is a vector of length $\mathrm{poly}(n) \cdot N$.



- $D$: Fix $q = \mathrm{poly}(n)$ and $m = O(n \log q)$, and $N = \mathrm{poly}(n)$. Pick $b_1^*, \ldots, b_m^*$ randomly from $\{0, 1/q, \ldots, q - 1/q, 1\}^n$.



## Intuition

Rapid Hand-waving.