Theorem: (0-1)-Permanent is \#P-complete

Proof:

\[
\text{\#Exact-3-Cover} \leq \text{\#W-BI-MATCH} \leq \text{\#PERM}
\]

1. \[
\text{\#(0,1)-d-Perm} \leq \text{\#(0,1)-(0-1)-Perm}
\]

Input: Set \( X = \{ x_1, \ldots, x_n \} \) & collection \( Y \subseteq \binom{X}{3} \)

Output: \( \# \) of \( Z \subseteq Y \) s.t. each \( i \in X \) is covered exactly once.

We showed \( \odot \) last class.

\[\text{\#W-BI-MATCH}:\]

- **Input:** bipartite \( G \) with edge weights
- **Output:** \( \sum_{M \in Z} w(M) = \sum_{M \in Z} \sum_{e \in M} w(e) \)
  
  where \( Z \) = all matchings of \( G \).

\[\text{\#Perm}:\]

- **Input:** bipartite \( G \) with edge weights
- **Output:** \( \sum_{P \in \mathcal{P}} w(P) \) where \( \mathcal{P} \) = all perfect matchings of \( G \).
Let's prove 2:

For integer \( k \geq 0 \), let

\[
\#W-k-BI-MATCH = \sum_{M \in Z : \left| M \right| = k} w(M) = \text{total weight of all matchings of size } k.
\]

We'll show: \( \#W-k-BI-MATCH \leq \#PERM \)

Then summing over \( k \) we get: \( \#W-BI-MATCH \leq \#PERM \).

Take bipartite \( G = (L \cup R, E) \) where \( |L| = l \), \( |R| = r \).

Let \( k \) be integer where \( 0 \leq k \leq \min\{l, r\} \).

Form \( G' \): Take \( G \),

- add \( v_1, \ldots, v_k \) to \( L \) connected to all of \( R \)
- add \( w_1, \ldots, w_k \) to \( R \) connected to all of \( L \).

Each \( M \in Z(G) \) of size \( \left| M \right| = k \) corresponds \( (l-k)! (r-k)! \) to perfect matchings of \( G' \).

Hence, \( \#PM \) of \( G' = \#W-k-BI-MATCH \times (l-k)! (r-k)! \).
Proof of (3):

\[ \# \text{(0,1)-Perm} \leq \# \text{(0,1)(l-1)-Perm} \]

Input: bipartite G with edge weights \( w \) where \( \leq 2 \)

Output: \( \sum_{P \in \mathcal{P}} w(P) \)

Gadget \( F_k \):

\( F_k \) has \( k \) perfect matchings

& 1 matching in \( N(u,v) \)

Take matrix \( A \) corresponding to input \( G \).

We want \( \text{per}(A) \).

Choose value \( \alpha \) we want to eliminate, \( \alpha \neq 5, 13 \)

Replace all \( \alpha \) by a variable \( x \) then

\( \text{per}(A) = \text{Polynomial } p(x) \) of degree \( \leq n \) in \( x \).
Instead: in $G$ replace each edge $(u,v)$ of weight $\alpha$ by a gadget $F_k$, for a parameter $k$.

Then: $\text{Per}(A) = p(k) = \text{polynomial in } k$.

# of non-$1$ entries went $\downarrow$ by $1$.

To evaluate $p(\alpha)$, evaluate $p(k)$ at $k \in \{0, 1, \ldots, n^2\}$ (just change size of gadget).

This yields $p(k)$ at $n+1$ different points.

Since degree of $p(k) \leq n$ then the value of $p(k)$ at $n+1$ points defines $p(k)$, i.e., we can get the coefficients of $p(k)$ by Gaussian elimination. So interpolate to get $p(k)$ at $k=\alpha$.\[\]
Can't hope for efficient exact counting algorithm for many counting problems, such as perfect matchings or all matchings

Instead: aim for approximation algorithm.

FPRAS: fully-polynomial randomized approximation scheme.

For \( \#(0,1)\)-Perm:

Given input \( G=(V,E) \) & error parameter \( \epsilon > 0 \), an FPRAS produces \( \text{OUT} \) s.t.

\[
\Pr \left( (1-\epsilon)|P| \leq \text{OUT} \leq (1+\epsilon)|P| \right) \geq \frac{3}{4}
\]

in time \( \text{poly}(n, \frac{1}{\epsilon}) \).

How to boost success probability?

Given \( \delta > 0 \).

Want to succeed with prob. \( \geq 1-\delta \).
Run FPRAS \( k = \frac{100 \ln\left(\frac{2}{\delta}\right)}{\delta} \) times &
get outputs \( y_1, \ldots, y_k \).
Let \( z = \text{median} (y_1, \ldots, y_k) \).
Output \( z \).

Analysis: Let \( X_i = \begin{cases} 1 & \text{if } y_i \in (1 \pm \epsilon) \eta \\ 0 & \text{otherwise} \end{cases} \)
\& \( X = \sum_{i=1}^{k} X_i \).

Note \( \mathbb{E}[X] \geq \frac{3}{4} k \).

Chernoff's bound:

Let \( X_1, \ldots, X_m \) be iid 50,15 r.v.'s where \( p = \mathbb{E}[X] \).
\& \( X = \sum_{i=1}^{m} X_i \).
Let \( \mu = \mathbb{E}[X] = mp \).

\[ \Pr (|X - \mu| > \epsilon \mu) \leq 2e^{-\epsilon^2 \mu / 2} \]
\[
\Pr(Z \leq (1+\epsilon)P) \leq \Pr(X < \frac{k}{2}) \\
\leq \Pr(|X - E[X]| > \frac{k}{4}) \\
\leq 2e^{-\frac{k^2}{46}} \\
\leq \delta.
\]

Hence to get error prob \( \leq \delta \),

takes \( O(\log(1/\delta)) \) time.

Approximate sampler:

\text{FPAUS}: fully-poly almost uniform sampler

Given \( G = (V, E), E_T \) & \( \delta > 0 \)

\text{FPAUS} generates \( P \in P \) from distribution \( \pi \) on \( \Theta \) where \( \Delta_{TV}(\mu, \pi) \leq \delta \)

in time poly \( (n, \log(1/\delta)) \)

where \( \mu = \text{uniform}(\Theta) \).
for μ & π on Ω

\[ d_{TV}(μ, π) = \frac{1}{2} \sum_{x \in Ω} |μ(x) - π(x)| \]

= \max_{S \subseteq Ω} μ(S) - π(S)

FPRAS ⇔ FPAUS (for self-reducible Problems [TVV])

Exact counter ➔ Exact-sampler