9^{TH} TUTORIAL ON RANDOMIZED ALGORITHMS

Counting matchings.

1. *Parity of perfect matchings.* Show an algorithm that given a bipartite graph G (partites consisting of the same number of vertices) determines if the number of perfect matchings is even or odd.

2. *Fraction of approximations.* We say that $\hat{x}$ is an ε-approximation of x iff

$$(1-\varepsilon)x \leq \hat{x} \leq (1+\varepsilon)x$$

Show that for $\varepsilon < 1/2$, if we have ε-approximation $\hat{s}$ of a number s and ε-approximation $\hat{t}$ of a number t, then $\hat{s}/\hat{t}$ is an 4ε-approximation of s/t. (It's sufficient to prove the upper bound as the lower bound is very similar.)

3. *Product of approximations.* Let $\varepsilon > 0$ be fixed. Find a suitable choice of $\overline{\varepsilon}$ such that if we take $\overline{\varepsilon}$-approximations $(\hat{a}_i)_{i=1}^n$ of numbers $(a_i)_{i=1}^n$, then $\prod_{i=1}^n \hat{a}_i$ is an ε-approximation of $\prod_{i=1}^n a_i$. (It's sufficient to prove the upper bound as the lower bound is very similar.)

4. *Main course: Counting matchings.* Let $G = (U \cup V, E)$ be a bipartite graph where $|U| = |V| = n$ and $\delta(G) > n/2$. We define:

$m_k =$ the number of matchings of size k in G, and

$r_k = m_k/m_{k-1} =$ the fraction of the # of k-matchings to the # of $k-1$-matchings.

Let $\alpha \geq 1$ be a real number such that $1/\alpha \leq r_k \leq \alpha$; for bipartite graphs with $\delta(G) > n/2$, it holds that $\alpha \leq n^2$. Pick $N = n^7\alpha$ elements from $M_k \cup M_{k-1}$ independently uniformly at random (approximately uniform generation covered in the lecture). Set $\hat{r_k}$ to the fraction of observed k-matchings to $(k-1)$-matchings. Show that

$$\left(1 - 1/n^3\right) r_k \leq \hat{r_k} \leq \left(1 + 1/n^3\right) r_k$$

with probability at least $1 - \exp(-n)$. (Hint: use the Estimator theorem from the lecture.)

Then show why accurate approximations of r_k's are useful for estimating the number of perfect matchings.

5. *Bonus: polynomial-time interactive protocol for permanent.* Show that permanent is in IP. We say that a language $L \subseteq \{0,1\}^*$ is in IP if

- The verifier V gets a word $w \in \{0,1\}^*$, works in polynomial time in $|w|$ and can use random bits.
- The verifier V can communicate with the prover P (which is computationally unbounded).
- We say that $L \in IP$ if there is a prover P and a verifier V such that:
  - Completeness: for each $w \in L$ we have
  $$\Pr[V(w) \text{ accepts the proof of } P] \geq 2/3$$
  - Soundness: for any $x \notin L$ and any prover Q we have
  $$\Pr[V(x) \text{ accepts the proof of } Q] \leq 1/3$$

Our goal is to show that the decision problem whether or not $\mathrm{perm}(A) = k$ for a given matrix $A \in \{0,1\}^{n \times n}$ and $k \in \mathbb{N}$ is in IP.