COS 594: Advanced Algorithms and Complexity **Spring 1999**

Assignment #2

Due: Monday, March 8 *Sanjeev Arora*

Suggested reading (for lectures 4,5,6): Spielman's Lecture Notes 7, 17, 18. My Barbados lecture notes upto page 10.

Problems (from lectures 4, 5, 6):

1. *(Randomized logspace)* Defining randomized logspace computations is subtle. The following definition of a randomized logspace machine is the "correct" one for several reasons.

   - Its workspace contains $\log n$ cells; the input is given on a separate read-only tape.
   - It can flip a coin, but if it wants to store the results of coin-flips, it has to store them on its work-tape.
   - Regardless of the results of its coin-flips, it must always stop in time $2^{O(\log n)}$ (i.e. it isn't allowed to go into an infinite loop).

   Now one can define the classes **RL**, **BP-L** analogously to **RP** and **BPP**.

   (i) Show that these classes remain unchanged if we assume that the random bits obtained from coin-flips are given on a separate read-only tape, to which the machine has *one-way access*. (A brief reason will suffice!)

   (ii) Show that **BP-L** $\subseteq$ **P**. (A hint: First think why **RL** $\subseteq$ **P**.)

   (iii) Now suppose the model is changed so that the machine has *two-way* access to the tape with random bits. Can you still show the results in (ii)? *(Okay, this is an open question)*

   (iv) Can the proof in class for **BPP** $\subseteq \Sigma_2^p$ also be used to show

   $$\textbf{BP-L} \subseteq \Sigma_2 - \textbf{SPACE}(\log n)?$$

2. (**A note on the permanent:**) In this question, $A = (a_{ij})$ will always be an $n \times n$ matrix with entries in a finite field $\mathcal{F}$ of p elements. Assume $p = 2^{poly(n)}$. Let $A^{i,j}$ be the sub-matrix of A obtained by deleting the ith row and the jth column. The *permanent* is a function on A defined as

   $$\begin{aligned} perm(A) &= \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)} \\ &= \sum_{i=1}^{n} a_{1i} \cdot perm(A^{1,i}) \end{aligned}$$

(S_n is the set of all permutations on n elements.) The definition of the permanent shows that it is *downward self-reducible*, since computing it on $n \times n$ matrices can be reduced to computing it on $(n-1) \times (n-1)$ matrices. The definition of the permanent may appear to be similar to that of the *determinant*, but its computation is believed to be much harder. Valiant ('79) showed that computing the permanent is $\sharp$**P**-complete. Toda ('89) showed that **PH** is contained in P^{perm}, where perm is an oracle for the permanent. We define

$$\text{PERM} = \{(M, k) : M \text{ is a matrix and } k = perm(M)\}.$$

Toda's theorem implies that if PERM $\in$ **BPP** then **PH** $= \Sigma_2^p$. In fact, if PERM $\in AM[2]$, then **co-NP** $\subseteq$ **NP**/*poly*, which implies that **PH** $= \Sigma_3^p$.

The goal of this exercise is to show that the permanent is likely to be hard to compute for almost all matrices. Namely, if some polynomial-time algorithm $\mathcal{T}_1$ correctly computes the permanent of a fraction $q = \frac{4n^2}{\sqrt{p}}$ of the $n \times n$ matrices over $\mathcal{F}$ (recall that $|\mathcal{F}| = p$), then **PH** $= \Sigma_3^p$. We'll show this by showing that the existence of T_1 implies that PERM $\in AM[2]$.

(a) Show Lipton's result, namely, if some deterministic algorithm computes $perm(A)$ correctly for a fraction $1 - \frac{1}{3n}$ of the matrices over $\mathcal{F}$, then PERM $\in$ **BPP**.
(Hint: For any $n \times n$ matrix R, $perm(A + x \cdot R)$ is a polynomial in x of degree n. In particular, setting $x = 0$ gives $perm(A)$. You should pick R randomly, i.e. pick its entries uniformly and independently from $\mathcal{F}$.)

(b) **(i).** Show that there is a unique $(n-1) \times (n-1)$ matrix $B(x)$, whose entries are polynomials in x of degree $n-1$, and which satisfies

$$B(i) = A^{1,i} \qquad i = \{1, 2, \ldots n\}.$$

(ii). (LFKN'90) Show that PERM $\in AM[poly]$. (*Hint: $perm(B(x))$ is a polynomial in x of degree $(n-1)^2$.*)

(c) **(i).** Your protocol in for PERM probably runs in n rounds. Show that if the verifier in $b(ii)$ asks all its questions in 1 round, then the probability of accepting an input $(M, k) \notin$ PERM can go up to 1.

(iii). In this question the probabilities and expectations will always be over the choices of $(n-1) \times (n-1)$ matrices R_1 and R_2, which are chosen randomly. Let $D(x)$ be the unique matrix whose entries are polynomials in x of degree $n+1$, and which satisfies

$$D(i) = A^{1,i} \qquad i = \{1, 2, \ldots, n\}$$

and

$$D(n+1) = R_1$$

$$D(n+2) = R_2.$$

Show that if M_1, M_2 are arbitrary fixed matrices and α_1, α_2 are distinct elements of $\mathcal{F} \setminus \{1, \ldots, n\}$, then

(a). $\Pr[D(\alpha_1) = M_1] = \Pr[D(\alpha_1) = M_2]$

(b). $\Pr[D(\alpha_1) = M_1 \text{ and } D(\alpha_2) = M_2] = \Pr[D(\alpha_1) = M_1] \cdot \Pr[D(\alpha_2) = M_2]$.

(c). Let $Y =$ Number of values of x for which $\mathcal{T}_1$ answers correctly on $D(x)$. Show that $E[Y] \geq qp - n$ and

$$\Pr[Y < \frac{qp}{2}] \leq \frac{8}{qp}.$$

(iv). (*Combinatorial Lemma: Nice!*) Let Ω be some set. Let $A_1, A_2, \ldots, A_k$ be subsets of Ω such that and $\forall i: \ |A_i| \geq \beta \cdot |\Omega|$ and $\forall i \neq j: \ |A_i \cap A_j| \leq \alpha \cdot |\Omega|$. Then show that k must satisfy $\beta k - \alpha^2 \binom{k}{2} \leq 1$.

(v). Now notice that in (*iii*), $perm(D(x))$ is a polynomial in x of degree $(n+1)(n-1)$. Let T_1 be a procedure that computes the permanent correctly for a fraction q of all $n \times n$ matrices.
Show that there are at most $\frac{4}{q}$ polynomials in x of degree $(n+1)(n-1)$ whose value coincides with $\mathcal{T}_1(D(x))$ for at least $\frac{qp}{2}$ values of $x \in \mathcal{F}$.

(vi). Now show that the verifier of question $b(ii)$ can use $\mathcal{T}_1$ to carry out the protocol correctly in one round. (*Hint: How can part (iv) and (v) be used to restrict the prover's options? Use the Goldwasser-Sipser Set Lowerbound Protocol.*)