E0 224: Computational Complexity Theory
Indian Institute of Science
Assignment 2

Due date: Nov 2, 2017 **Total points: 60**

1. (**4 marks**) Show that there is a decidable language in $\mathsf{P/poly}$ that is not in P.

2. (**7 marks**) Prove that logspace uniform NC^1 is contained in L.

3. (**9 marks**) Show that we can add two n bit numbers using a bounded fan-in boolean circuit of depth $O(\log n)$ and size $n^{O(1)}$. Such a circuit has $n+1$ output gates.

4. (**10 marks**)

   (a) (**3 marks**) Show that there exists a boolean function $g : \{0,1\}^m \to \{0,1\}$ that requires a circuit of size $\Omega(\frac{2^m}{m})$ to compute it.

   (b) (**7 marks**) A boolean formula is a circuit in which the fan out of every node is at most 1. Let $\mathcal{F}$ be a boolean formula, such that $\text{size}(\mathcal{F}) = s$. Show that there exists a formula $\mathcal{F}'$ of size $s^{O(1)}$ such that $\mathcal{F}$ and $\mathcal{F}'$ compute the same boolean function and $\text{depth}(\mathcal{F}') = O(\log s)$.

5. (**9 marks**) Give a randomized algorithm that takes input two $n \times n$ matrices A and B with integer entries and does the following: If A and B are similar then with high probability the algorithm outputs an $n \times n$ invertible matrix C with rational entries such that $CAC^{-1} = B$; otherwise it outputs 'A not similar to B'. Ensure that your algorithm runs in polynomial time.

6. (**5 marks**) The class probabilistic poly time (PP) is defined as follows: $L \subseteq \{0,1\}^*$ is in PP if there is a probabilistic polynomial time Turing machine M such that

$$\Pr[M(x) = L(x)] > \frac{1}{2}.$$

   (a) (**3 points**) Show that $\mathsf{NP} \subseteq \mathsf{PP}$.

   (b) (**2 points**) Show that if $\mathsf{BPP} = \mathsf{PP}$ then PH collapses.

7. **(4 points)** Prove that $\mathsf{BPP} = \mathsf{RP}$ if and only if $\mathsf{BPP} = \mathsf{ZPP}$.

8. **(12 points)** Let $\mathbb{F}_p$ be a finite field of size p (a prime), and $\mathbb{F}_p^{n\times n}$ be the set of all $n \times n$ matrices with entries from $\mathbb{F}_p$. Assume $p > n$. The permanent of a matrix $M = (m_{ij})_{i,j\in[n]} \in \mathbb{F}_p^{n\times n}$, denoted by $\text{Perm}_n(M)$, is defined as

$$\text{Perm}_n(M) = \sum_{\sigma \in S_n} \prod_{i\in[n]} m_{i\sigma(i)}.$$

   Suppose $\mathcal{A}$ is an algorithm that on input $M \in \mathbb{F}_p^{n\times n}$ outputs $\text{Perm}_n(M)$ correctly for all but $\frac{1}{n^3}$ fraction of input matrices in $\mathbb{F}_p^{n\times n}$. Using $\mathcal{A}$ as a subroutine, design an algorithm $\mathcal{B}$ that outputs $\text{Perm}_n(M)$ correctly on *every* input $M \in \mathbb{F}_p^{n\times n}$, with probability at least $1 - \frac{1}{2^n}$. The running time of $\mathcal{B}$ (modulo subroutine calls to $\mathcal{A}$) should be polynomial in $\log p$ and n.