CS 221 Computational Complexity **Due: Friday March 2, 2018 at 8pm**

Problem Set 3

Instructor: Madhu Sudan *TA: Preetum Nakkiran*

1. **Sparse Problems.**
   A *sparse problem* is one where, for each $n \in \mathbb{N}$, the number of accepting inputs of length n is bounded by a polynomial in n.

   Show that if a sparse problem is NP-complete, then the polynomial hierarchy collapses to the second level (i.e. $PH = NP$).

   For extra credit (i.e., optional, but no discussions/collaborations/scouring-the-web allowed): Show that if a sparse problem is NP-complete then NP = P.

2. **Search BPP.**
   Recall the notion of promise-BPP-search alluded to in Lecture 8. Roughly, promise-BPP-search is the class of problems with probabilistic search, and verification routines. Formally, the following promise problem is complete for promise-BPP-search:

   - **Input:** A "search" circuit $C : \{0,1\}^k \to \{0,1\}^n$ and a verification circuit $V : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ (the search circuit searches for a solution, and the verification verifies a solution, both probabilistically).
   - **Promises:**
   $$\forall x \in \{0,1\}^n, \Pr_y[V(x,y) = 1] \notin (1/3, 2/3)$$
   $$\text{and } \Pr_z\left[\Pr_y[V(C(z),y) = 1] \geq 2/3\right] \geq 2/3$$
   - **Task:** Output x such that $\Pr_y[V(x,y) = 1] \geq 2/3$.

   (a) With the above definition, show that promise-BPP-search can be amplified, so that $\Pr_y[V(x,y) = 1] \notin (2^{-\Omega n}, 1 - 2^{-\Omega n})$ and $\Pr_z\left[\Pr_y[V(C(z),y) = 1] \geq 1 - 2^{-\Omega n}\right] \geq 1 - 2^{-\Omega n}$.

   (b) Show that if promise-BPP = P, then promise-BPP-search = P. *(Hint: Find z such that $C(z)$ is a solution to the search problem. Try to find z one bit at a time by defining appropriate promise-BPP problems.)*

3. **Relaxed BPL**
   Give a formal definition of BPL, the class of problems that can be solved with randomized logspace algorithms and polynomial running time.

   The rest of this question - motivates the explicit running time restriction. Consider a relaxation of BPL where we do not restrict the runtime of the machine to be polynomial. Show that Relaxed-BPL contains NL.

4. **Derandomizing BPL**
   Show that BPL is in L^2.

   (Hint: Consider the technique used in Problem 4 on Pset1, to show $NL \subseteq L^2$)