COS 487: Theory of Computation	**Fall 1997**

Assignment #5

Due: Monday, December 8 — *Sanjeev Arora*

Suggested reading (for lectures 18–22): Sipser Chapter 7, 8.1 -8.5, 9.1.

Problems (from lectures upto Dec 1)

General Note: A pointer to an NP-completeness reduction in some book does not constitute a valid answer.

1. Problem 7.14 in Sipser.
2. Problem 7.15 in Sipser.
3. Problem 7.26 in Sipser.
4. Problem 7.34 in Sipser.
5. Problem 8.16 in Sipser.
6. Problem 8.19 in Sipser.
7. Recall that a language is PSPACE-hard if every language in PSPACE is polynomial-time reducible to it. Suppose we try to define this concept using a new kind of reduction, the *linear space* reduction. This is defined as follows. A *linear space reducer* is a turing machine that has three tapes: a read-only input tape, a read-write work tape, and a write-only output tape. When the input string has size n, the machine is allowed to use $O(n)$ cells on its work tape. It prints its output (which is a string) on the output tape. We say that language A is *linear space reducible to language* B (denoted $A \leq_{ls} B$) if there is a linear space transducer M such that for each $x \in \Sigma^*$, the output of M on x, denoted $M(x)$, satisfies: $M(x) \in B \Longleftrightarrow x \in A$.

   We say that a language is *PSPACE-tough* if every language in PSPACE is linear space reducible to it.

   (i) If $A \leq_{ls} B$ and $B \in$ PSPACE, then can we conclude that $A \in$ PSPACE?

   (ii) Describe a language in SPACE($\sqrt{n}$) that is PSPACE-tough.

   (iii) Assume P $\neq$ PSPACE. Describe a PSPACE-tough language that is not PSPACE-hard. (Hint: Can you solve part (ii) even if the $\sqrt{n}$ is replaced by some smaller function?)