Theory Qualifying Exam, Spring 2006.

Choose any 6 of the following 8 questions.
Complexity: Problem 1:

Let $\mathcal{H}$ be a family of 2-universal hash functions mapping $\{0, 1\}^n$ to $\{0, 1\}^m$. Let $S \subseteq \{0, 1\}^n$ be a set of size at least $2^m$. Prove that

$$\Pr_{h \in \mathcal{H}} [h(S) \neq \{0, 1\}^m] \leq \frac{2^{2m+1}}{|S|}$$

Use this to design a randomized polynomial time algorithm with SAT oracle that computes the number of satisfying assignments to a SAT formula within $(1 \pm \epsilon)$ multiplicative error, where $\epsilon > 0$ is an arbitrarily small constant.
Complexity: Problem 2:
Show that if there is a $\text{CO} - \text{NP}$-Complete language in $\text{AM}$ then $\text{PH}$ collapses to the second level.
Algorithms: Problem 3: (Maximum coverage)

Given a universal set $U$ of $n$ elements, with nonnegative weights specified, a collection of subsets of $U$, $S_1, \ldots, S_l$, and an integer $k$, pick $k$ sets so as to maximize the weight of elements covered.

Show that the obvious algorithm, of greedily picking the best set in each iteration until $k$ sets are picked, achieves an approximation factor of

$$1 - \left(1 - \frac{1}{k}\right)^k > 1 - \frac{1}{e}.$$
Algorithms: Problem 4:

A \((d, c, \alpha)\)-expander is a graph \(G = (V, E)\) where each node has degree at most \(d\), and every subset \(S \subseteq V\) with at most \(cn\) nodes has \(|N(S)| \geq \alpha|S|\), where \(N(S)\) is the set of neighbors of points in \(S\).

Starting with a set \(V\) of \(n\) nodes, add a random matching between the vertices thus: (a) choose a random permutation \(v_1, v_2, ..., v_n\) of the nodes, and (b) add the edges \((i, v_i)\) for all \(i\). (We may have parallel edges and self-loops; that is fine.) Repeat this process \(d = 600\) times.

Prove that \(G = (V, E)\) is a \((2d, \frac{7}{20}, \frac{3}{2})\)-expander with probability at least 1/2. (Hint: what is the probability that some set \(S\) with \(|S| \leq cn\) does not expand?)
Cryptography: Problem 5:

Symmetric encryption with a deck of cards. (From the lecture notes of Bellare and Rogaway) Alice shuffles a deck of cards and deals it all out to herself and Bob (each of them gets half of the 52 cards). Alice now wishes to send a secret message $M$ to Bob by saying something aloud. Eavesdropper Eve is listening in: she hears everything Alice says (but Eve can’t see the cards).

Part A. Suppose Alice’s message $M$ is a string of 48-bits. Describe how Alice can communicate $M$ to Bob in such a way that Eve will have no information about what is $M$.

Part B. Now suppose Alice’s message $M$ is 49 bits. Prove that there exists no protocol which allows Alice to communicate $M$ to Bob in such a way that Eve will have no information about $M$.

What does it mean that Eve learns nothing about $M$? We refer to the definition of Shannon or perfect secrecy.
Cryptography: Problem 6:

Assume that there exists an efficiently computable and invertible block cipher that is a pseudorandom permutation secure against chosen-plaintext attacks (PRP-CPA secure). Show that there exists an efficiently computable and invertible block cipher that is PRP-CPA secure but NOT a pseudorandom permutation secure against chosen-ciphertext attacks (not PRP-CCA secure). PRP-CCA security is sometimes called strong PRP security.
Cryptography: Problem 7:

Let $f : \{0, 1\}^n \to \{0, 1\}^{\ell(n)}$ be a one-way function. Let $\mathcal{H} = \{ h : \{0, 1\}^{\ell(n)} \to \{0, 1\}^{2^n} \}$ be a family of pairwise independent hash functions. Define a function $g$ as

$$g(x, y, h) \overset{\text{def}}{=} (h(f(x)), h),$$

where $x, y \in \{0, 1\}^n$ and $h \in \mathcal{H}$. Prove that $g$ is a length-preserving one-way function, and moreover $g$ essentially preserves the security of $f$, that is, if there is a PPT $A$ that inverts $g$ with probability $\varepsilon(n)$ (over random choices of $(x, y, h)$ and coin tosses of $A$), then there is a PPT $B$ that inverts $f$ with probability at least $\varepsilon(n) - 2^{-n}$. 
Cryptography: Problem 8:

Assume that one-way functions exist. Give a zero-knowledge proof of knowledge (ZKPOK) for the language CLIQUE, and prove its completeness, (knowledge) soundness (by giving a knowledge extractor), and zero-knowledge (by giving a simulator).

Note: You are required to give a ZKPOK system directly for CLIQUE, and are not allowed to reduce CLIQUE to another NP-Complete problem (such as 3-Coloring or Hamiltonian-Cycle) whose ZKPOK systems are well known.