When showing polynomial-time (or log-space) mapping reduction $A \leq_p B$, make sure you follow the following order: (1) briefly explain the intuition; (2) precisely and compactly define the reduction (also using some clear picture unless the reduction is absolutely obvious); (3) show the reduction is polynomial time (log space) in the size of $A$; (4) argue that if $x \in A$ then $f(x) \in B$; (5) argue that if $x \not\in A$, then $f(x) \not\in B$. When showing that a problem $B$ is $C$-complete (for some $C$, you should: (1) show that $B \in C$; (2) show that $A \leq B$ (using polytime or log-space reduction as stated above) for some known C-complete $A$. Forgetting one of these steps will result in point loss.

Problem 4-1 (NFA minimization is hard) 20 points

Recall that the problem of finding the smallest (in terms of the number of states) DFA equivalent to a given DFA was reasonably easy (in $P$). The objective of this problem is to show that a similar problem for NFA’s is much harder. Namely, let MIN-NFA be the decision problem, where given an NFA $N$ and an integer $n$, one has to determine if there exists an NFA $N'$ with at most $n$ states which is equivalent to $N$: $L(N) = L(N')$. Show that MIN-NFA is $PSPACE$-complete. 

(Hint: The harder part is to show it belongs to $PSPACE$. For completeness, reduce from $\text{ALL}_{\text{NFA}}$. The reduction is very simple, but be careful not to make it too simple.)

Problem 4-2 (Unary Languages and PSPACE) 15 points

Recall that if a unary language is NP-complete, then $P = NP$. Show a similar statement for PSPACE: if a unary language is PSPACE-complete, then $P = PSPACE$. Make sure you stress where you proof differs from that about NP. You are allowed to use the following lemma without proof.

Lemma 1 Let $B$ be a tree with $M$ nodes having depth at most $d$. Assume each node of $B$ is marked with a label from $\{1 \ldots p\}$ with the following property: if two nodes have the same label, one of them must be an ancestor of the other in $B$. Then we must have $M \leq dp$.

Problem 4-3 (The Cat, the Mouse and the Hole) 15 points

Consider the following game which seems to be PSPACE-complete on the first glance, but only turns out to be P-complete. The game is played on an undirected graph $G$ by a cat and a mouse. Initially both players occupy some node of $G$ and take turns (the cat being first) to move to a neighboring node in $G$. The objective of the mouse is to get to the “hole” – a special node of the graph $G$. The objective of the cat is to “catch the mouse” – occupy the same node with the mouse. Mouse wins if she gets to the hole first. Cat wins if she catches the mouse. If mouse/cat configuration repeats, it’s a draw. Show that the problem of determining if the cat wins (given $G$, position of the hole and the initial positions of cat/mouse) is in $P$.

(Hint: use dynamic programming.)
Problem 4-4 (P-complete Problem) 15 points

Recall the emptiness language for context-free grammars: $E_{CFG} = \{ G \mid G \text{ is a CFG and } L(G) = \emptyset \}$. Show that $E_{CFG}$ is P-complete.

(Hint: Reduce from monotone circuit value.)

Problem 4-5 (Bipartiteness) 15 points

Let $\text{Bipartite} = \{ G \mid G \text{ is a bipartite graph} \}$. Show that $\text{Bipartite} \in \text{NL}$.

Problem 4-6 (Matrix Multiplication) 20 points

Recall, a product of two $n \times n$ boolean matrices $A$ and $B$ is another $n \times n$ boolean matrix $C$ where $C_{i,j} = \bigvee_{k=1}^{n} (A_{i,k} \land B_{k,j})$.

(a) Show that matrix multiplication can be computed in logarithmic space.

(b) Using repeated squaring and reusing space, argue that computing the $p$-th power of $A$ can be done in space $O(\log n \log p)$.

(c) Using the above, give an alternative proof that $\text{NL} \subseteq \text{SPACE}(\log^2 n)$.

Problem 4-7 (Parentheses and brackets) 20 points

Let $B$ be a language of properly nested parantheses and brackets. For example $([[]([[])])]) \in B$, but $[()] \notin B$. Show that $B \in L$.

Hint: start with just balanced parenthesis, and see how you can reuse space from there.

Problem 4-8 (Polylogarithmic Space) 15 points

Let $\text{polyL} = \bigcup_{i \geq 1} \text{SPACE}(\log^i n)$ be the set of languages decided in polylogarithmic space.

(a) Show that $\text{NL} \neq \text{polyL}$.

(b) Show that $\text{P} \neq \text{polyL}$.

(Hint: Use the fact that $\text{P}$ has complete problems under log-space reductions.)

Problem 4-9 (A Major Circuit) 10 points

Let $\text{Majority}(x_1, \ldots, x_n)$ be 1 if $\sum x_i \geq n/2$ and be 0 otherwise. Show that $\text{Majority}$ can be computed by a circuit of size $O(n)$.

(Hint: Use divide and conquer and your knowledge of recurrence equations.)