Notes, book, etc. are not allowed, except that a "crib sheet" (one side of one piece of paper, per our rules) is allowed.

Except where otherwise noted, you must justify your answers. Correct answers with no justification may receive little credit. Incorrect or incomplete answers that display insight often receive partial credit. For example, if you know what form an argument should take, but you can't fill in the details, then at least show me the form.

You may cite material (definitions, theorems, algorithms, etc.) discussed in class, the assigned homework, or the assigned textbook sections. If you wish to use other material, then you must develop it first.

It is understood that clear, efficient, concise solutions are usually favored over confused, inefficient, or verbose solutions, and hence may earn more points.

If you feel that a problem is ambiguously worded, then ask for clarification. If the problem is still unclear, then explain your interpretation in your solution. Never interpret a problem in a way that renders it trivial.

You have 150 minutes. Good luck. :)

A.A. Here are six complexity classes. Wherever possible, draw a hooked arrow "$A \hookrightarrow B$" to indicate that A is a subset of B. Annotate each arrow with a brief (four words or fewer?) justification. For tidiness, do *not* include arrows that follow from the others by transitivity.

$EXPTIME$ NP

$EXPSPACE$ P

$NPSPACE$ $PSPACE$

A.B. In the diagram above, there should be a smallest complexity class (meaning that it's a subset of all others) and a largest (that is a superset of all others). Draw a Venn diagram that shows how those two complexity classes, and the five computability classes below, are related.

RegLs RecLs DecLs CoRecLs CFLs

B. In this problem, assume that the input alphabets for all TMs and NTMs are $\Sigma = \{0, 1\}$. Here is a nondeterministic recognizer for ${EMPTY_{TM}}^c$. On input $\langle M \rangle$:

1. Scan right across the tape to the first $\textvisiblespace$ (blank).
2. Nondeterministically select an arbitrary string x over Σ, and write $\#x\#$ to the tape.
3. Run M on x, and output whatever M outputs.

Draw a chunk of NTM state diagram showing in full detail how step 2 could be implemented.

C. Suppose that $A \leq_P B$ and $B \in NP$. Prove that $A \in NP$.

D. See below for a particular context-free grammar G in Chomsky normal form. We have discussed a polynomial-time algorithm for deciding whether a given string $w \in L(G)$. Execute the algorithm on $w = aabab$. Show the table that is constructed.

$S \to AB$

$A \to AA \mid a$

$B \to a \mid b$

E.A. In the Cook-Levin proof that SAT is NP-hard, a 2×3 "window" is an important concept. For the Turing machine partially described below, give two examples of legal windows.

$Q = \{q_0, q_1, q_2, q_{\text{acc}}, q_{\text{rej}}, \ldots\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sqcup\}$,

$\delta(q_0, 0) = (q_1, 1, \mathcal{R})$, $\delta(q_1, 0) = (q_{\text{acc}}, 0, \mathcal{R})$, $\delta(q_2, 0) = (q_1, 0, \mathcal{R})$,

$\delta(q_0, 1) = (q_2, 1, \mathcal{L})$, $\delta(q_1, 1) = (q_0, 1, \mathcal{L})$, $\delta(q_2, 1) = (q_{\text{rej}}, 1, \mathcal{L})$, $\ldots$

E.B. For the same Turing machine, give one example of an illegal window.

F. Suppose that a language A is decided by an NTM N in space $s_N(n)$. Give an upper bound for how much time is needed to decide A on a TM.