Open book, open notes, closed neighbors, 170 minutes. The exam has six problems and totals 240 pts., subdivided as shown. Show your work—this may help for partial credit. Please write in the exam books only.

**Notation:** The names of complexity classes are either completely standard or explained in the questions themselves—although these two classes are not expressly referenced, it may still help you to clarify \( E = \text{DTIME}[2^{O(n)}] \) and \( \text{EXP} = \text{DTIME}[2^{2^{O(1)}}] \). The alphabet \( \Sigma \) over which languages are encoded is immaterial; you are always welcome to consider \( \Sigma = \{0, 1\} \) or \( \Sigma = \{0, 1, \#\} \). The alphabet \( \Gamma \) used as the worktape alphabet of Turing machines may, however, be much larger. The tupling notation \( \langle x, y \rangle \) or \( \langle x, y, z \rangle \) may be considered either as representing the strings \( x\#y \) and \( x\#y\#z \) or as the application of pairing functions as in some other sources; in no problem does the difference really matter. As usual the notation “\( \#0(x) \)” means “the number of 0s in the string \( x \)” and so on. Always “\( x \leq y \)” refers to the standard “lex” order on strings, which is the same as the usual order on corresponding natural numbers.

You may cite any major relevant theorem or fact or definition covered in the course without needing to give a justification (unless one is specifically asked for). Some parts refer to “current knowledge” in complexity theory.

**Time Guideline:** Up to 45 minutes for Problem (1), 20 minutes for (2), 30 for (3), 15 minutes for (4), 20 minutes for (5), 20 minutes for (6), 20 for (7).

(1) (40 pts.)

Classify each of the following languages \( L_1, \ldots, L_{10} \) according to the classification on the next page. Please write your answers in this form: if \( L_{11} \) were the language of the Halting Problem, you would write “11. i” or “11. (i)” , perhaps adding the words “c.e. but undecidable” to guard against silly errors. The classes and languages are on the next page. **No justifications are needed**, but may help for partial credit. There is a unique best answer for each language, and some answer(s) may be unused. Unless otherwise specified, “\( M \)” or “\( M_1 \)” etc. refers to a deterministic Turing machine.
(a) regular;
(b) in deterministic logspace but not regular;
(c) in NL and not known to be—or believed not to be—in deterministic logspace;
(d) in P and not known to be—or believed not to be—in NL;
(e) in NP and strongly believed not to belong to co-NP;
(f) in PSPACE and strongly believed not to belong to NP;
(g) decidable but known to be not in PSPACE;
(h) c.e. but not decidable;
(i) co-c.e. but not c.e.
(j) neither c.e. nor co-c.e.

The languages are:

1. $L_1 = \{ \langle M, x \rangle : M$ is a DTM that halts and rejects $x \}$.  
2. $L_2 = \{ \langle M \rangle :$ on every input $x$, $M$ halts and rejects $x \}$.  
3. $L_3 = \{ \langle M \rangle :$ on every input $x$, $M$ does not halt $\}$.  
4. $L_4 = \{ \phi : \phi$ is a 2CNF formula in $n$ variables and for all $a \in \{0,1\}^n, \phi(a) = \text{false} \}$.  
5. $L_5 = \{ \phi : \phi$ is a 3CNF formula in $n$ variables and for all $a \in \{0,1\}^n, \phi(a) = \text{false} \}$.  
6. $L_6 = \{ \phi : \phi$ is a 3CNF formula in $n$ variables and $\phi(a_{\oplus}) = \text{false}$, where $a_{\oplus}$ is the assignment that sets $x_i = 1$ if $i$ is odd, $x_i = 0$ if $i$ is even. $\}$.  
7. $L_7 = \{ x \in \{0,1\}^* : \#0(x)$ is a multiple of $7 \}$.  
8. $L_8 = \{ x \in \{0,1\}^* : \#0(x)$ is a power of $7 \}$.  
9. $L_9 = \{ \langle G \rangle :$ the complement of $G$ is 4-colorable $\}$.  
10. $L_{10} = \{ \langle C, x \rangle : C$ is a Boolean circuit with $n = |x|$ input gates and $C(x) = \text{false} \}$.  

(2) 29 pts. total

Recall the three-qubit quantum circuit shown to have entanglement on assignment 6: Hadamard gate on line 1 followed by CNOT with control on line 1 and target on line 2, followed by Hadamard again on line 1. We can regard this circuit programmatically as $\text{H 1 CNOT 1 2 H 1}$. Recall it has the matrix

$$A = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{bmatrix}$$

Now suppose we have $\text{H 2 CNOT 2 1 H 2}$. Derive the matrix $B$ of this circuit. Then say what $AB$ is (this is ordinary multiplication, not tensor product) and say whether it entangles the basis input $e_{00}$.

(3) (42 pts. total)

(3) Show that the following decision problem is NP-complete.

**Instance:** A collection $S_1, ..., S_r$ of subsets of a finite set $U$.

**Question:** Can $U$ be partitioned into two subsets $V$ and $W$ such that every set $S_i$ contains at least one element of $V$ and one element of $W$?

A reduction from 3SAT is expected, but you may use any of the forms of 3SAT discussed in the course.

(4) (30 pts.) True-False with reasons.

Please write out the words true and false in full, for 3 points, and for the other 2 points, write a relevant justification (need not be a full proof, and should be brief).

(a) A computation that uses exponential time must use exponential space.

(b) The language TQBF of true quantified Boolean sentences logspace many-one reduces to its complement via the function $g(\phi) = \neg \phi$.

(c) On current knowledge, TQBF is hard for co-NP under logspace many-one reductions.

(d) At least one of NL $\neq$ P and P $\neq$ PSPACE must be true.

(e) There is a language $B \in \text{NP} \cap \text{co-NP}$ such that a polynomial-time program with $B$ as oracle can factor integers.

(f) The intersection of a regular language and a decidable language must belong to P.
(5) 30 pts.
Define the shuffle $x@y$ of two equal-length strings $x = x_1x_2\cdots x_n$ and $y = y_1y_2\cdots y_n$ to be the string $x_1y_1x_2y_2\cdots x_ny_n$, which has length $2n$. For example, $\lambda@\lambda = \lambda$, $000@111 = 010101$, $00@01 = 0001$, and $01@01 = 0011$ not $0101$. Over the alphabet $\Sigma = \{0, 1\}$, prove via the Myhill-Nerode Theorem that the following language is non-regular:
\[ L = \{ x@y : |x| = |y| \land \#1(x) = \#1(y) \}. \]

(6) (18 + 18 = 36 pts.)
Consider the following decision problem:

**Cycling DFA**
**Instance:** A DFA $M = (Q, \Sigma, \delta, s, F)$ and a string $x \in \Sigma^n$ where $Q = \{1, 2, \ldots, n\}$.
**Question:** Does $M$ on input $x$ visit every one of its states and end up back at its start state $s$?

(a) Sketch a deterministic Turing machine $M$ that decides this problem in $O(\log n)$ space. It is enough to diagram the worktapes of $M$ and say what information each one maintains while sketching the algorithm in pseudocode.

(b) Now sketch a faster algorithm that uses linear space. You may use the fact that Turing machines can execute mergesort at full efficiency. Estimate its running time and compare with your answer to (a).

(7) (24 + 9 = 33 pts.)
Prove that the language $\{ (M, 0^n) : L(M) \cap \{0, 1\}^n = \emptyset \}$ is not c.e. Here it is understood that $M$ stands for a deterministic Turing machine whose input alphabet is $\{0, 1\}$, and that $n$ is a non-negative integer ($n = 0$ allowed). Also answer and justify briefly: is the language co-c.e.?

End of Exam.