6.045J/18.400J: Automata, Computability and Complexity	Nancy Lynch

Quiz 3 Information

April 20, 2005 — *Vinod Vaikuntanathan*

1 General Information on Quiz 3

Quiz 3, on time complexity, will take place during class (11:00AM-12:30PM) on Wednesday, April 27. The quiz will cover Chapter 7 of Sipser's text, as well as the time hierarchy theorem from Section 9.1, and the material on oracle Turing machine time complexity in Section 9.2. You are responsible for material covered in lectures 16-21 (April 4 to April 25). You may bring your copy of Sipser's text with you to class to use during the quiz. You may also bring a two-sided sheet of handwritten notes.

2 Material You Should Know for the Quiz

The following is a list of important topics and ideas we covered during the time complexity portion of the course.

- **Asymptotic Function Notation.** You should be comfortable using the notation $O(t(n))$, $o(t(n)$, $\Theta(t(n))$, and $\Omega(t(n))$.
- **Time Complexity.** You should understand how to measure the running time of a Turing machine. You should understand the notion of a time-bounded complexity class, and should be comfortable with the notation $TIME(t(n))$.
- **The Class P.** You should know the definition of the complexity class P. You should understand why complexity theorists use P as their model for efficient computation. You should be able to use the definition of P to prove basic properties of this complexity class. You should know examples of languages in P and be able to sketch proofs that they are in P.
- **The Time Hierarchy Theorem.** You should be familiar with the statement of the time hierarchy theorem given in class. You should understand the basic ideas used in the proof of this theorem.
- **Nondeterministic Time-Bounded Turing Machines.** You should know what it means for a nondeterministic Turing machine (NTM) to be a decider. You should know how to measure time for a nondeterministic decider TM. You should know a reasonable bound on the time needed for a deterministic TM to simulate an NTM. You should be familiar with the notation $NTIME(t(n))$.
- **The Class NP.** You should know both of the definitions of NP that were given in class—based on nondeterministic Turing machines and based on certificates and verifiers. You should know why these two definitions are equivalent. You should know many problems in NP, including 3COLOR, CLIQUE, VERTEX-COVER, HAM-CYCLE, TSP, SUBSET-SUM,..., and should be able to show that they are in NP. You should be able to use the definition of NP to prove basic properties of this complexity class. You should understand the question of whether P = NP, but you are not required to know the answer.
- **The Class co-NP.** You should know the definition of the class co-NP. You should understand the relationship between the classes P, NP and co-NP.
- **Polynomial-Time Reducibility.** You should know the definition of polynomial-time reducibility. You should be able to prove basic properties of this reducibility. You should be able to prove that one language is polynomial time reducible to another language.
- **NP-Completeness**. You should know the definition of NP-completeness and you should understand how to prove languages are NP-complete. You should be familiar with the main ideas of the proof of the Cook-Levin theorem.

- **NP-Complete Problems.** You should be familiar with the NP-Complete problems that we have discussed in class (SAT, CNF-SAT, 3SAT, CLIQUE, VERTEX-COVER, HAM-PATH, HAM-CYCLE, TRAVELING-SALESMAN, SUBSET-SUM, PARTITION, MULTIRPOCESSOR-SCHEDULING, 3-DIMENSIONAL MATCHING, 3-EXACT-COVER). You should understand the basic ideas of the proofs that these are NP-complete.
- **Decision Problems vs. Optimization Problems.** You should understand the relationship between optimization problems and corresponding decision problems.
- **Higher-Order Time Complexity Classes.** You should know the definitions of the exponential time complexity classes, the relationships among these classes, and the relationships between these classes and the polynomial time complexity classes.
- **Oracle Time Complexity Classes.** You should know the definitions of the basic oracle Turing machine time complexity classes and the statements of the "relativized" results about P vs. NP.