Theory Comprehensive Exam

The comprehensive exam in theory will be drawn from the topic areas in three courses, ECS 120, ECS 122A,

and ECS 222A. About one third will be from ECS 120, about one third will be from ECS 122A, and

about one third will be from ECS 222A. The problems will require proofs as well as knowledge of results

and algorithms, and will require creative solutions as well as demonstration of knowledge about existing

methods and results. Below is an expanded list of topic areas that may be included in the exam.

ECS 120

1. Regular languages. Finite automata and the class of languages they define.

Closure properties of regular languages. Regular Expression. The pumping lemma and using it

to prove that certain languages are not regular. Proofs required for all these topics.

2. Context-free languages. Grammars and pushdown automata. Understand the definitions, and

statement of the relationship of PDA and CFLs. Proofs not required for these topics.

3. Computability Theory

The Turing-machine model, the RAM model, and other equivalent models of effective computability.

The Church-Turing thesis. Universal Turing machines.

Decidable and undecidable problems. Language A_TM is not decidable.

Other examples of undecidable and unrecognizable languages. Reductions to prove

additional languages undecidable or unrecognizable. Proofs required for these topics.

Understanding many-one reductions is compulsory.

4.Complexity Theory

Time complexity. Precise definitions of P and NP.

Polynomial-time reducibility. NP-Completeness. Cook's Theorem.

Example reductions among NP-hard problems.

Understanding poynomial-time reductions is compulsory.

Exam based on material in textbook:

M. Sipser, "Introduction to the Theory of Computation", any edition.

ECS 122A

1. Precise definitions of what the "Complexity of algorithms" means. Asymptotic notation and the meaning of upper and lower bounds on complexity.

2. Divide and Conquer. Recursive methods, such as mergesort, quicksort and related algorithms. Time analysis by setting up and solving recurrence relations. The unwrapping method and the Master method for recurrence relation solution.

3. Greedy algorithms.

4. Dynamic Programming for shortest path problems and other examples of DP. Dykstra's shortest path algorithm. Backtracking in dynamic programming.

5. Graph algorithms for Minimum Spanning Trees, graph biconnectivity, Ford-Fulkerson algorithm,

Max-Flow-Min-Cut theorem.

6. Branch-and-bound technique.

7. Approximation methods.

8. An intuitive view of NP-completeness (less formal than for ECS 120) and examples of various NP-complete problems.

9. Some randomized algorithms.

ECS 222A

All of the topics in ECS 122A are also included in the list for ECS 222A, but with more advanced algorithms and harder problems.

For example, analysis of randomized quicksort. Other randomized algorithms, such as for finding the max-cut in a graph. Fast matrix multiplication or fast integer multiplication. Dynamic programming for sequence alignment and edit distance, or RNA folding. Applications of network flow, such as bipartite matching or finding disjoint paths, More NP-completeness reductions. More involved approximation methods and results.

The parts of the exam drawn from ECS 122A and ECS 222A are based on material in Cormen, Leiserson, Rivest and Stein, "Introduction to Algorithms", any edition; and Kleinberg and Tardos, "Algorithm Design", 2005.