Class Presentation
Due: October 19, 2010

Below, there is a list of complexity classes. For each complexity class, I have mentioned one problem that is known to be in the class. There are also some references you might find helpful to read. (This is meant as a starting point; you might have to read other material in order to prepare your presentation.)

Each student taking the course will choose a different one of the following complexity classes as the topic for a brief (20-minute max) presentation to the class during the week of October 19. The presentation should include a formal definition of the class (probably preceded by an informal definition), a bit of motivation about why it is a useful class to think about, and an explanation of why the example problem mentioned below is a member of the class. (Note: in some cases, the problems mentioned are complete for the class, but you don't have to show that; you just have to show that the problem is in the class.)

To claim a problem, send me email with a ranking your top choices. (You may give a ranking of up to five problems.) I will process email messages in the order I receive them. I will give you the highest ranked question on your list that is not already claimed by someone else. If there is no such question I will respond asking you to give a ranking of the remaining questions.

1. APX: optimization problems with polynomial time constant-factor approximation algorithms.
Example problem: Euclidean Travelling Salesman Problem.

- Cormen, Leiserson, Rivest, Stein. *Introduction to Algorithms*, 2nd edition, chapter 35.2.
- Papadimitriou, Chapter 13.
- N. Christofides, Worst-case analysis of a new heuristic for the travelling salesman problem. In *Algorithms and Complexity: New Directions and Recent Results*, 1976.

2. #P: functions that compute the number of accepting computations of a polynomial time nondeterministic Turing machine.
Example problem: Computing permanent of a binary matrix.

- Papadimitriou, Section 18.1.
- L. G. Valiant, The complexity of computing the permanent, *Theoretical Computer Science*, 8, pp. 189–201, 1979.

3. PSPACE: problems that can be solved using polynomial space.
Example problem: Geography game.

- Papadimitriou, Section 19.1
- Thomas J. Schäfer, On the complexity of some two-person perfect-information games, *Journal of Computer and System Sciences*, 16, pp. 185–225, 1978.

4. $\Pi_2\mathbf{P}$: Problems solvable by Π_2-alternating polynomial-time Turing machines.
Example Problem: Minimum formula.

- Sipser, Section 10.3
- Ashok K. Chandra, Dexter C. Kozen and Larry J. Stockmeyer, Alternation, *Journal of the ACM*, 28(1), pp. 114–133, 1978.

5. RL: Problems that have randomized algorithms that run in logarithmic space.
Example Problem: reachability in an undirected graph.
Note: Reingold has shown reachability can even be solved deterministically in logarithmic space, but that is a much harder result.

- Papadimitriou, Section 16.3.
- Romas Aleliunas, Richard M. Karp, Richard J. Lipton, Laszlo Lovász and Charles Rackoff, Random walks, universal traversal sequences, and the complexity of maze problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 218–223, 1979.

6. SC: Steve's Class
Example Problems: all deterministic context-free languages.

- Stephen A. Cook, Deterministic CFL's are accepted simultaneously in polynomial time and log squared space, In *Proc. ACM Symposium on Theory of Computing*, pages 338-345, 1978.

7. BPP: problems that have polynomial time probabilistic algorithms which have a bounded probability of making a mistake.
Example Problem: PRIMES.
Note: This problem requires some number theory.

- Sipser, Section 10.2
- Michael O. Rabin, Probabilistic Algorithm for Testing Primality, *Journal of Number Theory*, 12, pp. 128–138, 1980.

8. Parity P: problems that can be decided by determining whether the number of accepting paths of an NP Turing machine is even or odd.
Example Problem: Graph automorphism
Note: This question requires some abstract algebra.

- Johannes Köbler, Uwe Schöning and Jacobo Torán, Graph Isomorphism is Low for PP, *Computational Complexity*, 2, pp. 301–330, 1992.