CS 525 - Computational Complexity

K. Subramani
LDCSEE,
West Virginia University,
Morgantown, WV
ksmani@csee.wvu.edu

1 General Information

1. Semester - Spring 2009.
2. Meeting Times: Tu-Th, 08:00-09:15 am.
   Location: 801, ESB-E.
3. Contact Information: 749 ESB, ksmani@csee.wvu.edu.
4. Office Hours: TR, 10:00-11:00 am.
5. Textbook - [Pap94] is the main text, although [CLRS01] is strongly recommended for supplementary reading.
6. URL - `http://www.csee.wvu.edu/~ksmani/courses/sp09/cc/cc.html`.
7. Assessment:
   (a) Homeworks (3) - There will be three Homework assignments; each assignment is worth 15% of your grade. Table (1) details the Homework schedule.

Assignment Date	Submission Date
01/29	02/12
02/19	03/05
04/2	04/17

Table 1: Homework Schedule

   (b) Presentations (2) - You will be required to present two topics which will be decided through discussions with the instructor. The first presentation will be worth 10 points and the second presentation will be worth 15 points.
   (c) Final - The final examination will be a take-home assignment. The questions will be posted on May 1 and need to be turned in by May 7, 5 pm. The final is worth 30% of your grade.
   (d) A maximum of 5 bonus points will be awarded for class performance.

8. Grade Boundaries:

Grade	Boundary
A	80 and up
B	$65 - 79$
C	$50 - 64$
D	$45 - 49$
F	$0 - 44$

Table 2: Grade Boundaries

9. Grading policy - If you have any questions about the grading, you must contact the intructor within two days of your paper being returned.
10. Makeup Policy - If for some reason, you are unable to attend a test or an exam, please meet me at the earliest and I will set an alternate date.
11. **Course Objectives** - The objectives of this course are as follows:
    (a) Introducing machine-independent notions of resource (time and space) analysis.
    (b) Rigorously establishing time and space bounds for selected problems.
    (c) Developing the notions of undecidability and completeness.
    (d) Developing the notion of complexity classes.
    (e) Categorizing problems into appropriate complexity classes.
12. **Learning Outcomes** - Upon successful completion of this course, students will be able to:
    (a) Recognize undecidability in a language.
    (b) Distinguish between complexity classes.
    (c) Appreciate the logical characterization of complexity classes.
    (d) Categorize problems into appropriate complexity classes.
    (e) Identify the possibility of intractability for a given problem.

2 Syllabus Sketch and Weekly Schedule

2.1 Algorithms - Machine Models

Three paradigmatic problems, viz., Graph Reachability, Maximum Flow and Traveling Salesman Problem, algorithmic efficiency, reductions. (1 Lecture).

2.1.1 Turing machines

Turing machine basics, Turing machines as algorithms, Multiple strings, Linear speedup, Space bounds, Random Access Machines, Nondeterministic machines. (1.5 Lectures).

2.1.2 Computability

Universal Turing machines, The Halting problem, Other undecidable problems. (1.5 Lectures).

2.2 Logic

2.2.1 Boolean Logic

Boolean expressions, Satisfiability and validity, Boolean Functions and Circuits. (1 Lecture).

2.2.2 First-order Logic

Syntax, Models, Valid expressions, Axioms and proofs, The completeness theorem, Consequences of the completeness theorem, Second-order logic. (3 Lectures).

2.2.3 Undecidability in Logic

Axioms for number theory, Complexity as a number-theoretic concept, Undecidability and incompleteness. (2 Lectures).

2.3 P and NP

2.3.1 Relations between Complexity Classes

Complexity classes, The hierarchy theorem, The reachability method. (2 Lectures).

2.3.2 Reductions and Completeness

Reductions, Completeness, Logical characterizations. (2 Lectures).

2.3.3 NP-completeness

Membership, Satisfiability variants, Graph-theoretic problems, Sets and numbers. (2 Lectures.)

2.3.4 coNP and function problems

`NP` and `coNP`, Primality, Function problems. (2 Lectures).

2.3.5 Randomized computation

Randomized algorithms, Randomized complexity classes, Circuit complexity. (2 Lectures).

2.3.6 Approximability

Approximation algorithms, Approximation and complexity, Inapproximability. (2 Lectures).

2.3.7 On P vs. NP

The map of `NP`, Isomorphism and density, Oracles, Monotone circuits. (2 Lectures).

2.4 Beyond NP

2.4.1 The polynomial hierarchy

Optimization problems, The hierarchy. (2 Lectures).

2.4.2 Computation that counts

The permanent, The class $\oplus$`P`. (1 Lecture).

2.4.3 Polynomial Space

Alternation and games, Games against nature and interactive protocols, Additional `PSPACE`-complete problems.
(2 Lectures).

I would like to reiterate that this is a sketch of the topics that we will be covering. For various reasons, I may choose to drop a mentioned topic or cover a new topic. In such cases, advance notice will be given.

3 Social Justice Statement

West Virginia University is committed to social justice. I concur with that commitment and expect to foster a nurturing learning environement, based upon open communication, mutual respect and non-discrimination. Our University does not discriminate on the basis of race, sex, age, disability, veteran status, religon, sexual orientation, color or national origin. Any suggestions to further such a positive and open environment in this class will be appreciated and given serious consideration. If you are a person with a disability and anticipate needing any type of accommodation in order to participate in this class, please advise me of the same and make appropriate arrangments with Disability Services $(293-6700)$.

If you feel that you are being treated inappropriately or unfairly in any way, please feel free to bring your concerns to my attention; rest assured that doing so will not prejudice the grading process. In return, I expect you to behave professionally and ethically.

References

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press, 2001.

[Pap94] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, New York, 1994.