NC 34, Traveling Salesmen and Other Puzzles
Syllabus, Spring 2007, Block VII

Professor: Dr. Gretchen Ostheimer
Office: Weed 209
Phone: 463-6106
Email: cscgzo@hofstra.edu
Office hours: to be announced

1 Overview

In this class you will explore two famous problems in computer science: the Traveling Salesman Problem and the Euler Circuit Problem. For both of these problems we are given a map, and our goal is to use a computer to find a route that satisfies certain criteria, and to do so as quickly as possible. In the Traveling Salesman Problem, our goal is to find the shortest route that visits each location exactly once. In the Euler Circuit Problem our goal is to find a route that travels along each bit of road exactly once. Fast solutions have been found for the Euler Circuit Problem, but computer scientists still don’t know if there is a fast solution to the Traveling Salesman Problem. This latter question is perhaps the most famous unsolved problem in mathematics and computer science today.

What do we mean, exactly, by a “fast” solution? Why would one of these problems be so much more difficult to solve than the others? More generally, what are the inherent limitations of computing? Are there problems which can’t be solved at all using a computer? These are the central questions of this class.

All of these problems have important applications. In this class we will focus on applications related to transit planning.
2 Learning Objectives

Liberal Arts Objectives:

1. to learn to solve unfamiliar problems (without being taught how to solve them!);
2. to think clearly and analytically;
3. to work cooperatively;
4. to read closely;
5. to write and speak precisely;
6. to reflect on the role of the university and liberal arts education in society.

Computer Science Objectives:

1. to understand what is meant by problem specification, algorithm description and algorithm analysis;
2. to understand the classification of problems into tractable, intractable and undecidable, and to be able to provide examples of each type;
3. to appreciate the practical importance of this classification scheme;
4. to be able to use technical language in writing and speaking precisely and accurately about current unsolved problems in computer science;
5. for each of the following problems: the Halting Problem, the Traveling Salesman Problem, the Hamilton Circuit Problem, and the Euler Circuit Problem, to be able to classify the complexity of the problem and to be able to give a complete justification for that classification.

Programming Objectives:

1. to be able to run Python programs for solving classical graph theory problems;
2. to be able to import real-world data into a Python program;
3. to understand the syntax and semantics of simple Python statements;
4. to be able to write a simple Python program using
   (a) assignment and print statements,
   (b) function calls, with parameters and return statements,
(c) conditional statements,
(d) loops, both definite and indefinite,
(e) recursion.

5. to be able to modify a somewhat complex Python program in order to change the algorithm employed for solving the given problem.

To get a C in your programming work, you must demonstrate mastery of the objectives up to and including function calls and conditional statements. To get a B in your programming work, you must also demonstrate your ability to work with definite and indefinite loops. To get a B+, A- or A, in your programming work, you must make headway with recursion, and with modifying a somewhat complex Python program.

Mathematics Objectives:

1. to be able to use mathematical language and notation correctly, in the context of syntactically and semantically correct English;
2. to understand the role of definitions, examples and theorems in mathematics;
3. to be able to check a proof for correctness;
4. to be able to present a proof orally and answer questions about it;
5. to know the basic definitions of graph theory;
6. to understand the statements and proofs of simple theorems about paths in graphs.
3 Readings

   (a) Chapter 1, Pure Mathematics;
   (b) Chapter 2, Graphs;
   (c) Chapter 8, Euler Walks and Hamilton Walks.

   (a) Section 2.1 Algorithms;
   (b) Section 2.3 Complexity of Algorithms;
   (c) Section 8.5 Euler and Hamilton Paths;


It is recommended that you buy Trudeau and the Python book. Trudeau and Rosen will be on electronic reserve, and the Python book is available for free on-line at http://greenteapress.com/thinkpython/ if you don’t want a bound copy.

4 Graded Work

In each of the grading categories below, your grade will be based on two things: the extent to which you can demonstrate that you have achieved the learning objectives above by the end of the block, and the extent to which you have been consistently handing in work on time and according to the instructions. The expectation is that you will be devoting an average of 28 hours in addition to classtime to your work for this class per week, since you are expected to work for two hours outside of class for every hour in class. Everyone can get an A in this class, because everyone has the ability to work hard and to meet with me often. Every time we review your work together you will be given the opportunity to revise it according to my instructions, and to thereby nudge the quality up toward an A.

Please note that this grading system is highly non-standard. For example, I don’t expect anyone to hand in A level work until the very end of the semester, since the learning objectives are chosen in such a way that the entire block will be needed to achieve them. Also note that in the case of the programming work, which is worth 30% of your grade, the grade is based entirely on what you can demonstrate in scheduled meetings with me. If at any stage during the block you start to worry about your grade in the class, it is imperative that you come talk to me. We will look over your work together, and assess how you are doing vis-a-vis the learning objectives.
Programming Quizzes: Regular quizzes will be used to test your understanding of the syntax and semantics of Python assignment and print statements, function calls, conditionals and loops, and recursion.

Programming: You will be given a series of programming exercises designed to help you achieve the programming objectives above while reinforcing the computer science and mathematics objectives. Your grade for this section of the class will depend entirely on your ability to present this work to me in private meetings. In these meetings you will be asked to test your programs in front of me and to modify your programs to demonstrate that you thoroughly understand your own work. You must come to these meetings on time and well-prepared, since if you don’t, then you lose an opportunity to redo the work.

Lab Write Ups: You will write a lab report for each programming exercise. In this lab report you will describe the problem you were asked to solve, you will exhibit your solution and your tests of your solution. You will justify your choice of tests. These documents must be self-contained: the reader should not need to refer to the assignment to understand what you are trying to do. You can rewrite the labs as often as you like; the only stipulation is that you hand in your rewrites in private meetings so we can go over them together; be sure to schedule these meetings well ahead of time. In order to get an A on a lab write-up, it must be complete in the sense described above, it must have good sentence and paragraph structure, and it must be free of any spelling mistakes.

Group Work Write Ups: For each of the following problems: the Traveling Salesman Problem, the Hamilton Circuit Problem, the Euler Circuit Problem and the Halting Problem, you will write a report summarizing your findings. Each report will have the same format:

- one paragraph will describe the problem specification, defining any terms used;
- there will be one paragraph for each of the algorithms you learned for solving the problem, or you will state that no such algorithm exists and you will support that claim;
- for each algorithm above, there will be one paragraph giving the time complexity of the algorithm, and describing how that formula is obtained;
- if there is an open question from computer science related to this problem, there will be one paragraph describing the status of that question.

You can rewrite your reports as often as you like: the only stipulation is that you hand in your rewrites in private meetings so we can go over them together; be sure to schedule these meetings well ahead of time. In order to get an A on a group work write-up, it
must be complete in the sense described above, it must have good sentence and paragraph structure, and it must be free of any spelling mistakes.

**Definitions:** You are required to develop a stack of index cards containing the definitions for which you will be responsible. These definitions must consist of complete English sentences. It is recommended that you copy the definitions from a reliable source since at this stage in your career it will be too difficult to paraphrase a definition without losing precision.

**Definition, Proof Scramble and True-False Quizzes:** Three kinds of quizzes will be given to test your ability to read mathematics and to think creatively about it:

- **Definition quizzes,** in which you will be asked to give definitions from memory;
- **true-false quizzes;**
- **quizzes in which you will given a set of sentences that comprise a proof, and you will be asked to order them appropriately to make the argument.**

**Oral Presentations:** You will be asked to present a mathematical proof to the class without the use of notes, and to field questions of the following kinds: give an example that illustrates this theorem; give an example that illustrates why the hypotheses are necessary; what is the definition of this term? how does this line follow from that one? and so on.

**Paper:** Each of you will write one paper, with several revisions. The paper will be an ethnographic study of the community of mathematics researchers based entirely on field work and interviews. Everyone is required to attend the following workshop:

- **Languages, groups, complexity and cryptography on Wednesday, April 11.**
  
  www.math.stevens.edu/~melder/languages

  (We have class on Thursday, April 12, so you cannot attend that day.)

In addition, each of you must attend one of the following:

- **New York Group Theory Seminar Friday, April 13,** Graduate Center, CUNY. 365 Fifth Avenue at 34th Street.
  
  www.grouptheory.org/nyseminar.html
Meet for the tea that starts at 3:30pm in the Mathematics Lounge on the 4th Floor. You will need picture ID to get into the building. The security guards will ask for the room number where you are going. Take the elevators to the 4th Floor, look both ways and turn toward the hallway that has information monitors, go through those glass doors, turn left to a T, then left and the Mathematics Lounge is immediately on your right. The talk is in Room 5417 on the 5th Floor. We all wander up there together.

- AMS Sectional Meeting **Saturday and Sunday, April 14 and 15.**

  [www.ams.org/amsmtgs/2140_program.html](http://www.ams.org/amsmtgs/2140_program.html)

  I will be at the Special Session on Languages and Groups, so you might feel most comfortable there.
5 Grades

Grade Breakdown:

1. programming meetings 30%
2. tracing quizzes 15%
3. paper 15%
4. lab write ups 10%
5. group work write ups 10%
6. definition, true-false and proof scramble quizzes 10%
7. proof presentation 10%

Absence Policy: Attendance in class, at meetings and at events is mandatory. For each class or activity off-campus missed for unexcused reasons, your final grade will be reduced by one letter grade (that is, from A to B, and so on). In the event that you are unable to attend class or an activity off-campus due to illness or a university sanctioned event, you must speak with me immediately about the possibility of making up that requirement. If you fail to make up the required work or accumulate a total of four absences, whether excused or not, you will not be able to earn credit for this block course.

Academic Honesty: Students enrolled in this class implicitly promise to adhere to Hofstra’s policies regarding academic honesty. Whenever you consult anyone (another student in this class, another student not in this class, a tutor, an instructor, anyone) you must acknowledge the help you received in writing, either as a comment in your Python code, or as a separate section of your lab write up, your group work write up, or your paper. It is expected that every piece of work you hand in, including the programs you run for me in my office, will have an acknowledgments section. Students who are found to have violated their promise (either by cheating or by assisting another student in cheating) are given a 0 on the given assignment, are reported to the dean, and may, furthermore, receive an automatic F in the class.