Slide 1

Administrivia

- Final exam May 14 (Tuesday) at noon. Review sheet describing format and topics on Web.
- Reminders: Homework 7 due today, Homework 8 next Wednesday at 5pm. We need a "not accepted past" date for this and other homeworks. Next Friday (5/10) at 5pm?
- Solutions for homeworks and midterm (to be) available in hardcopy form. (I will put them in my mailbox in Halsell, the ones already distributed today, others no later than next Wednesday? I will also let you know if/when I get remaining homeworks graded.)
- I'm intending to have office hours next week but am not sure when. I will send mail when I know.
- Should there be a review session sometime?
- Extra-credit problem set possible if there is interest. ?

Slide 2

Quote of the Day/Week/?

- Mathematicians are like Frenchmen: whatever you say to them they translate into their own language, and forthwith it is something entirely different. (Attributed to Goethe.)

Slide 3

Recap — Course Goals

- For CS majors, learn math needed for later courses. Something of a "grab bag" of topics, but you probably *will* see some of this material again. For non-CS majors, introduction to some math you might otherwise not encounter.
- Increase "mathematical maturity" — in part, this is the ability to think logically, especially valuable to people in CS (also other science, engineering, math), but good for others too.
- A recurring theme is to take something that might be difficult to think through from first principles and turn it into a symbol-manipulation problem.

Slide 4

Topics and Why We Covered Them

- Formal logic:
  - Understanding connectives/tautologies related to simplifying boolean expressions, e.g., in programs.
  - Example of "formal system" — CS people will deal with others, e.g., formal grammars (basis for compilers, e.g.).
  - "Mathematical maturity".

  (Aside: Dr. Myers recommends that CS majors consider a symbolic logic course as one of your math electives. Students who take it seem to find it worthwhile.)

Slide 5

Topics and Why We Covered Them, Continued

- Proof techniques (direct proof, contraposition, proof by contradiction, proof by induction):
  - Background for courses that involve proofs.
  - "Mathematical maturity".
- Program correctness:
  - Another way to think about programs – even if not applied formally, E.g., "loop invariant" idea — recall problem with black/white marbles.

Slide 6

Topics and Why We Covered Them, Continued

- Recursion:
  - Recursive definitions used in theory courses.
  - Recursive algorithms sometimes easier to express than iterative equivalents (e.g., anything working with trees).
- Analysis of algorithms:
  - Simplified version, but gives background for discussions in programming classes.
- Recurrence relations:
  - Useful in doing "analysis of algorithms" on recursive algorithms.

Slide 7

Topics and Why We Covered Them, Continued

- Sets, counting, and probability:
  - General background. (Stuff about infinite sets is a tangent, but an interesting one?)
  - Often useful to know how many cases must be considered.
  - "Expected value" calculations (in optional reading) useful in doing analysis of algorithms for average case (rather than worst case, as we did before).

Slide 8

Topics and Why We Covered Them, Continued

- Relations and functions:
  - General background.
  - Background for formal study of relational databases. (Definitions of set operations needed here too.)

Slide 9

Minute Essay

- None — sign in.