Slide 1

Administrivia

- Final exam May 6 (Friday) at 2pm. Review sheet describing format and topics on Web.
- Things to review:
  - Class notes (online and your own), including minute essays (solutions in online notes).
  - Homework problems (including the "not to turn in"). Solutions to all homeworks available by Monday noon. Graded papers — as soon as I can.
  - Quizzes. (Solutions online.)
- No formal review session, but office hours today 1:30pm to 2:30pm and Monday 3pm to 5pm. (I'll probably also be around later Friday and earlier Monday. Stop by, or call/e-mail to find out if I'm around.)

Slide 2

Administrivia

- Reminder: Homework 8 due today at 5pm; accepted until noon Monday.
- Warning/caveat: I will be out of town from midday Tuesday noon to late Thursday. I will (try to) check e-mail and voice mail daily, though, so please e-mail and/or call if you have questions.

Slide 3

Administrivia

- Numeric grade is "points earned" divided by "maximum points" on the following:
  - Two exams (100 points and 200 points).
  - Quizzes (50 points — lowest score dropped).
  - Class attendance/participation — 50 points.
  - Homework — 220 points.
- Letter grades assigned "on a curve" (no attempt to fit a bell curve, but median numeric score is about a B-).
- There will also be a set of extra-credit problems. To be posted on Web Monday, due the following Tuesday. They *can only help your grade* — adding up to 30 points to "points earned" without changing "maximum points".

Slide 4

Recap — Course Goals

- Learn math needed for later courses. Something of a "grab bag" of topics, but you probably *will* see some of this material again.
- Increase "mathematical maturity" — in part, this is the ability to think logically, which can only help, both in programming and other CS courses.
- 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 5

Topics and Why We Covered Them

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

  (Aside: Dr. Myers recommends the symbolic logic course as one of your math electives. Students who take it seem to find it worthwhile.)
- Proof techniques (direct proof, contraposition, proof by contradition, proof by induction):
  - Background for courses that involve proofs.
  - "Mathematical maturity".

Slide 6

Topics and Why We Covered Them, Continued

- 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 7

Topics and Why We Covered Them, Continued

- Recursion:
  - Recursive definitions used in theory courses.
  - Recursive algorithms sometimes easier to write than iterative equivalents (e.g., tree traversal).
- 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 8

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.

Slide 9

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 10

Topics and Why We Covered Them, Continued

- Graphs and trees:
  - Abstraction behind some key data structures.
  - Trees you may have used already.
  - Many uses for graphs — serialization in Java, garbage collection, shortest path through a network, etc., etc.

Slide 11

Minute Essay

- None — quiz.