CSCI 1320 (Principles of Algorithm Design I), Fall 2008

Review for Final Exam

1 Format of the exam

The exam will be at the scheduled time for the course final, December 15 at 2pm. The exam will be about twice the length of the first exam (hence intended to take about two and a half hours), but you can use the full three-hour period if you like. You may use your textbook and any notes or papers you care to bring (with the exception of any materials from this course in previous years), but you may not use other books, each other's papers, or a calculator or computer. (If you think you might want to look at material from the course Web site, please bring printed copies. You can use the printers in Halsell.)

Questions will mostly be similar in format to the ones in quizzes and minute essays — some short-answer or multiple choice, some "what does this program do/print", some "write a program to do this task" — but probably somewhat longer and/or more difficult. Overall they will probably be similar in format, length, and difficulty to the questions on the midterm.

2 Topics to review

You are responsible for all material covered in class or in the assigned reading, but the focus will be on material presented or at least mentioned in class. (See the schedule for a list of assigned reading.) Also, the focus will be on material covered since the first exam, but there may be questions on earlier material as well, since (1) this exam is worth more points, and (2) the material is cumulative in nature. You should review in particular the following topics.

- Basics of how computers/programs work — source code, object code, executables; text editors and compilers.
- Basic Linux commands.
- Number systems — converting decimal to binary/octal/hexadecimal and vice versa, including fractions.
- Data representation — basic idea of how negative integers and floating point numbers are represented.
- Variables in C (types, declarations).
- Expressions and statements in C.
- Conditional execution in C.
- I/O in C (use of `scanf` and `printf`); character I/O and files.
- Functions in C — defining them, using them, using library functions.
- Loops in C; repetition and recursion.
- Arrays in C (1D only).

- Sorting and searching.
- Pointers in C, and how they're used to simulate pass-by-reference.
- Strings in C.
- Possibly a little about dynamic memory in C; a little about user-defined types in C, at the level of class discussion.