Topics in Programming: Competitive Programming (CS 224)
Fall 2014

Location: CS Department Lab, 16 Wing Technology Center
Time: Wednesday, 4:30pm–6:30pm
Instructor: Martin Allen
Office: Wing 210 (ext. 6810)
email: mallen <AT> cs <DOT> uwlax <DOT> edu

Course Description: Programming competitions are a great way to become a better programmer, and to have fun doing so. In a programming competition, teams work together under a time-limit to solve a set of problems, with the winners being those who correctly solve the most problems, in the shortest total time. This new course will introduce students to some of the techniques and problem types that pop up in many programming contests. If you are a relatively new programmer, or an experienced hacker, there is still a lot to learn from competitive programming.

Objectives for the Course: Students will learn a set of common algorithms, conceptual tools, and rules of thumb, intended to allow them to better solve programming problems of the sort found in programming competitions. Things you will learn how to do include, among other things:

- Applying various important algorithmic paradigms in practice, including (in no particular order):
  - Dynamic programming.
  - Divide-and-conquer.
  - Graph search and graph path.
  - Basic computational geometry.
- Identifying problem structure, including:
  - Combinatorial problems.
  - Graph problems (paths, matching, etc.).
- Programming using common tricks for efficiency, including:
  - When you might or might not want to use brute force.
  - Employing common handy data-structures.
- Understanding limits of language constructs, including working with values too large for basic data types.

Benefits of taking the course for students include:

- Practice in coding fundamentals.
- Practice in facing new problems and trying to find solutions in an open-ended manner.
• Increased knowledge of the common algorithms, mathematical approaches, and data-structures underlying many seemingly different problems.
• Learning why we need to have more than one tool in our tool-box for a certain type of computational problem.

Course Format and Expectations: The course will meet as a lab section for 2 hours, once a week. The instructor will provide some initial materials, problem examples, and discussion in the first half hour or so of a meeting. Students will then spend the remaining time trying to solve one or more practice problems. Difficulty of problems will vary, escalating over time as students become more experienced. Occasional meetings may take the form of team/individual ”mini-contests,” and student teams will have the (optional) opportunity to compete for fame and fortune in the upcoming ACM programming contest, held here at UW-L mid-semester.

Grading will be based solely on attendance, and on attempting problems. Some contest problems are very tricky, and part of the learning process is discovering through experience which sorts of problems are likely to be harder than others. It is not expected, nor required, that you will solve the majority of problems given, especially on your first try. A student who attends each week, and submits code attempts, will receive full credit for the course; grades will be adjusted downward for each day missed.

Please Note: Any student with a documented disability (e.g., physical, learning, psychiatric, vision, or hearing, etc.) who needs to arrange reasonable accommodations must contact the instructor and the Disability Resource Services office (165 Murphy Library, 785-6900) at the beginning of the semester. Students who are currently using the Disability Resource Services office will have a copy of a contract that verifies they are qualified students with disabilities who have documentation on file in the Disability Resource Services office. It is the student’s responsibility to communicate their needs with the instructor in a timely manner.