# Student Directed Seminar Proposal: Problem Solving with Algorithms

Kuan-Chieh Robert Tseng
Christopher Head

May 8, 2008

## 1 Course Content

For our student directed seminar, we intend to focus on taking the algorithms presented in CPSC classes — primarily 320 — understanding the algorithms at an intuitive level through class discussions involving pseudocode and diagrams, and then implementing the algorithms in code in one of a small set of programming languages.

As expanded further in the Rationale section, we feel that this course should appeal to a number of different classes of computer science students: those who share a passion for solving problems and learning about algorithms, those who intend to enter the software development industry and expect to potentially be in a situation where advanced algorithms are needed to solve real-world problems, and those who want to apply to one of the many companies for which a good practical understanding of algorithms is a prerequisite for entry and is often tested during interviews.

Students attending our seminar should have moderate programming skills in either C, C++, or Java and a good theoretical background in algorithms. We intend to enforce this by stating CPSC 320 as a prerequisite, though this prerequisite may be waived at our discretion if we feel an applicant possesses equivalent knowledge and skills from other sources. We do not at this time intend to require a particular grade level in either CPSC 320 or any other course, though due to the nature of the seminar we strongly recommend against anyone with poor results in CPSC 320 applying.

This seminar has been offered in a number of previous years as CPSC 490, and this will most likely remain true for our offering.

## 2 Course Structure

We intend to meet for 3 hours each week, most likely on a Monday/Wednesday/Friday schedule.

We intend to break our schedule into three types of classes: introductory lectures, sample problem presentations, and guest lectures.

### 2.1 Introductory Lectures

Each introductory lecture will be given by one of the course coordinators, and will review the theories behind a topic or algorithm as well as introduce any important practical knowledge for implementing the algorithm.

### 2.2 Sample Problem Presentations

Interleaved with the introductory lectures, the course coordinators will select a small number of problems in the topic and assign the problems to students. The students will then present pseudocode-level solutions to the problems in a subsequent class. Students other than those selected to present are encouraged to add their ideas to the discussion. Presenters will also locate other problems using similar algorithms to use as homework.

### 2.3 Guest Lectures

We intend to contact a small number of faculty doing research in algorithms to give presentations to the class. Depending on the topics presented, we may or may not include the material from guest lectures in homework (which

the coordinators would prepare) and on the quizzes.

# 3 Course Requirement and Evaluation

The tentative marking scheme is: assignments (40%), participation (30%), and quizzes (30%). These numbers may change in the first meeting.

## 3.1 Assignments

- There will be an assignment for each introduced topic. It will consist of 3 programming problems of varying difficulty.
- The assignment will be designed by the group of students who lead the discussion on the sample problems of the current topic.
- Students are encouraged to discuss the assignment, but must produce the final program by themselves. Plagiarism will not be tolerated.
- Assignment solutions will be marked via an automated judge. Partial solutions along with explanations may be submitted for partial marks.

## 3.2 Participation

- Class discussions are an integral part of the course. Students will have chances to both participate and lead discussions.
- Marks will be given using a peer evaluation system.

## 3.3 Quizzes

- There will be 2-3 quizzes throughout the term to test the students on the understanding of the topics presented.
- The quizzes will be on paper and will be designed by the coordinators.

# 4 Tentative Schedule

## 4.1 Introduction

- Input/Output in Java, C++

## 4.2 Graph Theory

- Shortest Path
- Minimum Spanning Tree
- Maximum Flow and Matching

## 4.3 Backtracking

- Branch and Bound
- Bidirectional Search

## 4.4 Dynamic Programming

- Memoization vs Iterative

### 4.5 Number Theory

- GCD, Primes, Euler's Totient Function
- Modular inverses, Chinese Remainder Theorem

## 5 Faculty Sponsor

Dr. Will Evans has kindly agreed to be the faculty sponsor for the course. Dr. Evans's research is closely related to this course. He has also taught CPSC 420 before.

## 6 Rationale for Offering the Course

While the computer science department offers courses on algorithms (CPSC 320, 420), these courses focus mostly on the theories and very little on practice. This course is meant to fill in the gap by giving students an opportunity to solve algorithmic problems, apply the algorithms learned in other courses, and actually implement the solutions. Therefore, this course will appeal to students who enjoyed the theoretical algorithmic courses and want to see how the theories can be applied.

One of the main goals of this course is to develop problem solving and critical analysis skills. These skills are the core of many real life programming projects. Our course will provide the students with valuable experience on handling real life projects without the additional complexity of managing a large software system. Furthermore, many companies use algorithmic problems during interviews to test candidates's skills, so students who are interested in getting jobs in the computer science industry will also find the course useful.

Finally, we believe that there are many students who share our passion in solving algorithmic problems. The format of the SDS provides the perfect setting for sharing our experiences in solving problems as well as passing on our interest in the subject.

## 7 Qualifications of the Coordinators

Kuan-Chieh Robert Tseng: $4^{\text{th}}$ year in Combined Honours Computer Science & Math
Student ID: 21215058
Related Experience:

- UTA for CPSC 221, 2007
- UBC ACM Programming Team Member, 2006–present
- Completed both CPSC 320 and 420

I have enjoyed solving programming problems since my first year. The process is fun and educational at the same time. By coordinating this seminar, I hope to share my passion with other students as well as gain experience in facilitating discussions.

Christopher Head: $4^{\text{th}}$ year in Computer Science Honours
Student ID: 23571052
Related Experience:

- UTA for CPSC 111, 2006
- UBC ACM Programming Team Member, 2005–present
- Completed CPSC 320
- Attended CPSC 490 as a non-coordinating student in 2006WT2

I started attending ACM team practices in September of my first year at UBC, and have been involved with the contest ever since. At the recommendation of another contestant, I applied to CPSC 490 in my second year, and found it both interesting and useful to my contest performance. In coordinating CPSC 490, I hope to pass on both skills and interest in the field of algorithmic problem solving to computer science students in general, and also possibly to some of the next generation of contestants.