
CS490: Problem Solving in Computer Science

Lecture 1: Introduction

Dustin Tseng

Mike Li

Wednesday January 4, 2006

- Administrativa

- Sample Problem

- Administrativa
- Sample Problem

Introduction

- ▶ Welcome to CS490 Problem Solving in Computer Science
- ▶ We will explore various useful algorithms/programming techniques and apply them to solve interesting problems
- ▶ CS490 is a Student Directed Seminar, as well as a full 3-credit 400-level CS course
- ▶ Sponsored by Dr. David Kirkpatrick
- ▶ 15 Students are permitted
- ▶ Course website: <http://www.ugrad.cs.ubc.ca/~cs490/>
- ▶ To register now → talk to Dustin or Mike

Objectives

- ▶ To achieve an in-depth understanding of a variety of algorithms
- ▶ To learn practical programming techniques
- ▶ To build up a personal code library
- ▶ To gain experience at presenting large topics
- ▶ To network with others who are similar interests

Format

- ▶ There is no final exam
- ▶ The course will be mainly seminars presented by students (20%)
- ▶ Each topics will have 2-4 homework questions assigned by presenters (20%)
- ▶ There will be three quizzes (60%)
- ▶ Final grade will depend on performance, attendance, and understanding of materials

Topics

- ▶ Dynamic Programming
- ▶ Brute Force Methods (backtrack, branch and bound, etc)
- ▶ Graphs::Shortest Path
- ▶ Graphs::Flow and Matching
- ▶ Number Theory (GCD, modular arithmetic, etc)
- ▶ Computational Geometry
- ▶ Feel free to bring up possible topics

Presentation

- ▶ You have the freedom of forming groups and picking topics
- ▶ Signup early to secure the topic you would like to present
- ▶ Should clearly state the problem and why it is of interest
- ▶ Introduce approaches to the problem, including both theory and actual code
- ▶ Evaluation will be performed by peers
- ▶ 70% for content
- ▶ 30% for style

Homework

- ▶ Homework questions should be representative of the topics
- ▶ Make sure the questions vary in terms of difficulties
- ▶ Penalties for assigning all easy or all hard questions
- ▶ Problem solvers will submit their solutions via online judging system
- ▶ Problem setters will present the solution after due date
- ▶ Discussion in groups is encouraged but coding and submission should be done individually

Quiz

- ▶ All three quizzes will be identical in format and have two components
- ▶ First part will be just like any in-class midterm
- ▶ You will be asked to present your ideas and logics to tackle the problems given
- ▶ Second part will be like a take-home exam similar to homework
- ▶ However, for the quizzes, you must do everything independently
- ▶ You are free to modify approaches you wrote down in the first part

Resource

- ▶ Electronic resources can be found on course website
- ▶ The “Big White Book”
- ▶ Coordinators (Dustin and Mike) are also here to help
- ▶ UBC ACM Team is another excellent source for help.
- ▶ We recommend everyone especially 3rd year students to join it.

- Administrativa

- Sample Problem

Problem Statement

Before Halloween, Bob wants to identify the friend with the best route, which will give both of them the maximum number of candies of all.

Given:

- ▶ an array of size 1000 containing numbers in $[-100, 100]$
- ▶ up to 50000 subsequences

Want:

- ▶ The subsequence with the maximum sum

Approaches

Brute Force

- ▶ Calculate each friend's benefit by summing over their routes
- ▶ Complexity: $O(n^2)$
- ▶ Array has size $n = 50,000 \rightarrow$ Takes too long!

Partial Sum

- ▶ Speed up the calculation for each friend by doing some precomputation

Partial Sum

- ▶ Original array
 - array = [5 3 6 2 1 5 6]
- ▶ Cumulative sum array
 - CumSum = [0 5 8 14 16 17 23 29]
- ▶ Find the sum over a range is now a single subtraction
- ▶ Complexity: $O(\text{sizeofarray}) + O(\text{numberofqueries})$

Smart Brute Force

- ▶ Input is sorted! (oops on my part...)
- ▶ Successive friends have similar start and end houses
- ▶ Calculate a friend's value by modifying the previous friend's value!
- ▶ Fast enough in this case
- ▶ Solutions are on the web under assignments

What Else?

- ▶ Course website: <http://www.ugrad.cs.ubc.ca/~cs490/>
- ▶ To register: talk to us now
- ▶ Everyone should send us a email with name, student number in order for us to setup the submission accounts
- ▶ Next class: STL, API, I/O