Course Outline

Course Staff

- LIC: Aleksandar Ignjatovic (ignjat) (56659)
- Lecturer: Ray Li (z3424810)
- Tutors: Declan McDonnell (z5163949), Michael Chen (z5118285), Sahan Fernando (z5113187)

Email me at `ray.li@student.unsw (complete the rest)`.

Thanks to Steven Fan and Tim Lambert for all their work in setting up this course and creating resources in previous years.

Thanks to Raveen de Silva and Greg Omelaenko for lecturing and creating resources in previous years.

Class Details

- Lectures
  - Tuesday, 12:00 - 14:00 in Ainsworth 202
  - Wednesday, 16:00 - 17:00 in Ainsworth 102
- Labs
  - Thursday, weeks 1–10, 12:00 - 15:00 in Drum Lab (K17 B8)
  - Thursday, weeks 1–10, 15:00 - 18:00 in Drum Lab (K17 B8)
  - Friday, weeks 1–10, 14:00 - 17:00 in Oboe Lab (J17 304)

Course Summary

1. Introduction
2. Data Structures
3. Graphs
4. Dynamic Programming
5. Network Flow
6. Strings
7. Mathematics

8. Computational Geometry

There is a tentative course schedule below.

Course Aims

This is a course designed to introduce advanced problem solving techniques to those who have already mastered the fundamentals of programming. A particular emphasis will be made on modifying existing algorithms to solve new problems in a novel way. Students will not only develop theoretical approaches to challenging problems, but also implement them.

Learning Outcomes

Upon completing this course, students will have mastery in applying various algorithms and data structures to solve competition-style problems, including implementation in C++.

Assumed Knowledge

- Students are assumed to have a thorough understanding of basic algorithms and data structures, as covered in COMP1511 and COMP2521.
- Students will benefit from having some familiarity with more advanced algorithms and data structures, as covered in COMP3121 or COMP3821.
- Students are expected to be proficient in C or C++, as all coding in this course will be in C++. A detailed understanding of C++ classes from an object-oriented perspective is not required, but we will make use of STL and structs with member functions.

Teaching Strategies

Lectures will introduce the theory behind the algorithms and data structures used in the course, as well as providing examples of applications to problems of varying difficulty.

Labs will provide students an opportunity to practice implementing these algorithms and receive guidance on how to solve problems using the concepts introduced in lectures.

Teaching Rationale

This course extends COMP3121/3821 to include applications of problem-solving techniques, in order to code solutions to problems, particularly in the style of programming contests. As such, the course trains students to solve problems in this field across a variety of topics, including under time constraints.

Assessment

Problem Sets	45%
Assignment	15%
Contests	16%
Final	24%
(Optional) Bonus Assignment	4%

Problem Sets

In all weeks except the last, a set of approximately 6 problems will be released, due 3 weeks later at midnight. The exact deadline will be listed in the problem set.

Each of the 9 sets will be worth 5% of the overall course marks, for a total of 45%.

To obtain full marks in a problem set, generally all but one or two problems needs to be completed. Additional problems completed award bonus marks towards the course. There is 1 bonus mark available each week. Specifics will be included in each week's problem sets.

Solutions must be implemented in C++.

Thanks to Jaehyun Park for advice on the problem sets.

Assignment

Students will be given a few problems of substantial difficulty but broken down into manageable subproblems with hints. The majority of the assignment will be automarked in a similar fashion to the problem sets with a small amount of marks for a brief solution writeup.

Students can work individually or in pairs.

Further information will be available when the assignment is released.

Contests

2 short contests (3 problems, 2.5 hours) will be held in labs in weeks TBA.

The contests are open book — any hard copy material can be brought in, including prewritten code. Lecture slides and C++ documentation will also be made available on the exam workstations.

Each contest will be worth 8% of the overall course mark, for a total of 16%.

Final Exam

The final exam will be a long contest (approx 9 problems, 5 hours), with the same conditions as above. Lecture notes and code will be provided in digital form, and you may bring any amount of written or printed material. Bringing your own digital resources or code is not allowed.

If it is necessary to break ties within the same course mark to obtain rankings, e.g. for first place, ties will be broken based on ranks obtained from the final exam according to ACM-ICPC scoring rules.

Bonus Assignment

There is an optional assignment for 4 bonus marks.

Form a group of 3 and pick a past ACM style contest. Attempt the contest and write up a short commentary on the problems your team solved. Then, with the help of the editorial if need be, solve and write a brief solution summary for most of the remaining problems.

Further information will be available later in the course.

Academic Honesty and Plagiarism

- Problem sets, contests, and the final exam must be completed individually. For problem sets, discussion of ideas is encouraged, but any code must be derived and written separately. Copying or purchasing code from external sources is also unacceptable.
- The assignments are to be completed in groups, but otherwise the same policy applies as above.
- The contests and final exam will be conducted under exam conditions.
- There are serious consequences for plagiarism. The university's policy can be found here. The School policies are located here.

Course Schedule

The following is a draft and is subject to change.

Week	2 hour lecture	1 hour lecture	Assessments
1	Intro	Data Structures I	
2	Data Structures I cont.	Graph Theory I	

Week	2 hour lecture	1 hour lecture	Assessments
3	Graph Theory I cont.	Graph Theory I with a focus on Trees	
4	Data Structures II	Data Structures II problems	
5	Dynamic Programming I	Divide and Conquer	Contest 1
6	Network Flow	Network Flow Problems	
7	Dynamic Programming II	Dynamic Programming II cont.	
8	Strings	Strings + Problems	Assignment 1
9	Computational Geometry	Mathematics	
10	Misc	Revision	Contest 2
14th Dec			Bonus Assignment

A problem set will be released at the start of every week except Week 10. Each will be due in 2 weeks, at the end of the week.

Note: 14th Dec is the last day of the final exam period.

Resources for Students

- The course website will contain all relevant resources, and there are no textbooks required for the course.
- The following textbooks are suitable for reference:
  - Any algorithms textbook, such as Introduction to Algorithms (Cormen, Leiserson, Rivest and Stein)
  - Programming Challenges (Skiena and Revilla)
  - Competitive Programming 3 (Halim). A free copy of Competitive Programming 1 can be found at CPBook
- Students may find the following online resources helpful:
  - CP-Algorithms (and the more comprehensive but Russian Emaxx)
  - Topcoder Tutorials

    - There are many helpful blogs, particularly on Codeforces (a partial list here). Some select choices:
        - Range Trees
        - Dynamic Programming On Trees
        - Geometry
        - Dynamic Programming Optimizations
- Further practice problems can be found in many places, e.g:
    - Codeforces
    - AtCoder
    - topcoder
    - codechef
    - USACO
- Students are encouraged to ask the lecturer or tutors for further resources.

COMP4128 2019t3

COMP4128 2019t3