Resources / Course Outline

Course Outline

Contents

Course Details

Course Code	COMP3121
Course Title	Algorithms and Programming Techniques
Convenor	Aleksandar Ignjatovic (/users/z9500319)
Admin	Aleksandar Ignjatovic (/users/z9500319)
Classes	**Lectures** : ...fill in times/locations of lectures... Timetable for all classes (/COMP3121/20T2/timetable)
Consultations	... fill in the times/locations of consultations...
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs3121/20T2/ (http://cse.unsw.edu.au/~cs3121/20T2/)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3121.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP3121.html)

Course Summary

Courses COMP 3121/3821/9101/9801 are about problem solving . Before you can start programming, you must first solve the problem at hand. Thus, we will **not** just go over a cookbook of some well-known algorithms which you have to memorise and just learn how to implement. Instead, we will learn how to design “from scratch” new algorithms for solving new problems, using various **design techniques** (greedy, dynamic programming, divide and conquer, etc). We will also learn how to estimate efficiency of algorithms. Extended courses COMP 3821/9801 are intended to provide ***additional depth.***

Among the well-known algorithms which use the same technique we will always choose to study those that are **most frequently used in practice** , in various fields, even if this requires a bit of extra work. For example, we will demonstrate the “divide and conquer” technique by studying the Fast Fourier Transform (FFT), which reigns in digital signal processing, in digital telecommunications and in countless other fields, including finance, for example.

Assumed Knowledge

Before commencing this course, students should:

- be familiar with basic algorithms and data structures
- be able to implement such algorithms and data structures

These are assumed to have been acquired in COMP2521

Student Learning Outcomes

After completing this course, students will:

1. Be able to design new algorithms for solving new problems, using various design techniques (greedy, dynamic programming, divide and conquer, etc)
2. Be able to estimate efficiency of algorithms and justify their correctness
3. Be able to demonstrate improved problem solving skills

This course contributes to the development of the following graduate capabilities:

Graduate Capability	**Acquired in**
Scholars capable of independent and collaborative enquiry, rigorous in their analysis, critique and reflection, and able to innovate by applying their knowledge and skills to the solution of novel as well as routine problems	?
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	?
Professionals capable of ethical, self- directed practice and independent lifelong learning	?
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	?

Teaching Strategies

- Lectures are dedicated to introduction of new concepts, illustrated by plenty of examples
- This course does not have tutorials; instead, more than 100 solved problems are made available to students as course material.
- Biweekly homework assignments are integral part of this course and serve to reinforce learned design techniques.

Teaching Rationale

This is a skill building course; thus, the emphasis is on problem solving through extensive practice

Student Conduct

The **Student Code of Conduct** (Information (https://student.unsw.edu.au/conduct) , Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)) sets out what the University expects from students as members of the UNSW community. As well as the learning, teaching and research environment, the University aims to provide an environment that enables students to achieve their full potential and to provide an experience consistent with the University's values and guiding principles. A condition of enrolment is that students *inform themselves* of the University's rules and policies affecting them, and conduct themselves accordingly.

In particular, students have the responsibility to observe standards of equity and respect in dealing with every member of the University community. This applies to all activities on UNSW premises and all external activities related to study and research. This includes behaviour in person as well as behaviour on social media, for example Facebook groups set up for the purpose of discussing UNSW courses or course work. Behaviour that is considered in breach of the Student Code Policy as discriminatory, sexually inappropriate, bullying, harassing, invading another's privacy or causing any person to fear for their personal safety is serious misconduct and can lead to severe penalties, including suspension or exclusion from UNSW.

If you have any concerns, you may raise them with your lecturer, or approach the School Ethics Officer (mailto:ethics-officer@cse.unsw.edu.au) , Grievance Officer (mailto:grievance-officer@cse.unsw.edu.au) , or one of the student representatives.

Plagiarism is defined as (https://student.unsw.edu.au/plagiarism) using the words or ideas of others and presenting them as your own. UNSW and CSE treat plagiarism as academic misconduct, which means that it carries penalties as severe as being excluded from further study at UNSW. There are several on-line sources to help you understand what plagiarism is and how it is dealt with at UNSW:

- Plagiarism and Academic Integrity (https://student.unsw.edu.au/plagiarism)
- UNSW Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism. In particular, you are also responsible that your assignment files are not accessible by anyone but you by setting the correct permissions in your CSE directory and code repository, if using. Note also that plagiarism includes paying or asking another person to do a piece of work for you and then submitting it as your own work.

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW staff and students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. Plagiarism at UNSW is defined as using the words or ideas of others and passing them off as your own.

If you haven't done so yet, please take the time to read the full text of

- UNSW's policy regarding academic honesty and plagiarism (https://student.unsw.edu.au/plagiarism)

The pages below describe the policies and procedures in more detail:

- Student Code Policy (https://www.gs.unsw.edu.au/policy/documents/studentcodepolicy.pdf)
- Student Misconduct Procedure (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- Plagiarism Policy Statement (https://www.gs.unsw.edu.au/policy/documents/plagiarismpolicy.pdf)
- Plagiarism Procedure (https://www.gs.unsw.edu.au/policy/documents/plagiarismprocedure.pdf)

You should also read the following page which describes your rights and responsibilities in the CSE context:

- Essential Advice for CSE Students (https://www.engineering.unsw.edu.au/computer-science-engineering/about-us/organisational-structure/student-services/policies/essential-advice-for-cse-students)

Assessment

Item	Topics	Due	Marks	Contributes to
Assignments 1 - 5	All topics	Weeks 3,5,7,9,11	50%	1,2,3
Final Exam	All topics	Exam period	50%	1,2,3

Course Schedule

Week	Lectures	Tutes	Labs	Assignments	Quizzes	Notes
1	Course intro and review material	-	-	assignment 1 released	-	-
2	Divide-and-Conquer	-	-	-	-	-
3	Recurrences, large integer multiplication	-	-	assignment 1 due assign. 2 released	-	-
4	Fast Fourier Transform	-	-	-	-	-
5	Greedy Strategy	-	-	assignment 2 due assign. 3 released	-	-
6	Review and Problem solving practice	-	-	-	-	-
7	Dynamic Programming	-	-	assignment 3 due assign. 4 released		-
8	Maximal Flow and Linear Programming	-	-	-	-	-
9	String matching	-	-	assignment 4 due assign. 5 released		-
10	Intractability	-	-	-	-	-
	-	-	-	assignment 5 due	-	-
	-	-	-	-	-	-
	-	-	-	-	-	-

Resources for Students

Texts and recommended readings:

- Kleinberg and Tardos: Algorithm Design
- Cormen, Leiserson, Rivest and Stein: Introduction to Algorithms
- Lecture notes and solved exercises available on Moodle

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

In the previous offering of this courses, students were generally quite happy with the course

Based on their comments, we have added more examples and solved problems

Resource created about a year ago (Tuesday 19 May 2020, 10:28:44 AM), last modified about a year ago (Tuesday 19 May 2020, 12:07:51 PM).

Comments

🔖 🔍 (/COMP3121/20T2/forums/search?forum_choice=resource/46269)

💬 (/COMP3121/20T2/forums/resource/46269)

There are no comments yet.