Course Outline

Contents

- Course Details
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Strategies
- Teaching Rationale
- Student Conduct
- Assessment
- Course Schedule
- Resources for Students
- Course Evaluation and Development

Course Details

Course Code	COMP9319
Course Title	Web Data Compression and Search
Lecturer	Raymond Wong (/users/z3002286)
Tutor	Haojie Huang (/users/z3409413)
Classes	**Lectures** : Thursday 18:00 - 21:00 Colombo Theatre A (K-B16-LG03) Timetable for all classes (/COMP9319/18s2/timetable)
Consultations	Haojie Huang: Wednesday 14:00 - 15:00 in K17 Level 2 - Consultation Level 2, K17 Raymond Wong: Thursday 17:00 - 18:00 in K17 Level 2 - Room 213 Haojie Huang: Friday 14:00 - 15:00 in K17 Level 2 - Consultation Level 2, K17
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs9319/ (https://webcms3.cse.unsw.edu.au/COMP9319/18s2)
Handbook Entry	http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9319.html (http://www.handbook.unsw.edu.au/postgraduate/courses/current/COMP9319.html)

Course Aims

As the amount of Web data increases, it is becoming vital to not only be able to search and retrieve this information quickly, but also to store it in a compact manner. This is especially important for mobile devices which are becoming increasingly popular. Without loss of generality, within this course, we assume Web data (excluding media content) will be in XML and its like (e.g., XHTML).

This course aims to introduce the concepts, theories, and algorithmic issues important to Web data compression and search. The course will also introduce the most recent development in various areas of Web data optimization topics, common practice, and its applications. The course is composed of the following parts:

- Adaptive coding, information theory
- Text compression (zip, gzip, bzip, etc)
- Burrows-Wheeler Transform and backward search
- XML compression
- Indexing
- Pattern matching and regular expression search
- Distributed querying
- Fast index construction
- Implementation

If time allows, we may cover optional topics such as: streaming algorithms, text analytics, Web data optimization for mobile devices.The lecture materials will be complemented by two programming assignments and numerous tutorial exercises.

Student Learning Outcomes

After completing this course, students will:

- have a good understanding of the fundamentals of text compression
- be introduced to advanced data compression techniques such as those based on Burrows Wheeler Transform
- have programming experience in Web data compression and optimization
- have a deep understanding of XML and selected XML processing and optimization techniques
- understand the advantages and disadvantages of data compression for Web search
- have a basic understanding of XML distributed query processing
- appreciate the past, present and future of data compression and Web data optimization

There are two programming assignments and one final exam in the course. Written exercises will also be provided and relate to the fundamentals of data compression. The programming assignments will relate to the applications of Web data compression and search techniques and/or their programming.

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	Exercises, Programming assignments
Entrepreneurial leaders capable of initiating and embracing innovation and change, as well as engaging and enabling others to contribute to change	Programming assignments, Classroom discussions
Professionals capable of ethical, self- directed practice and independent lifelong learning	Exercises, Provided readings
Global citizens who are culturally adept and capable of respecting diversity and acting in a socially just and responsible way	Case studies, Classroom discussions

Assumed Knowledge

Official prerequisite of this course is COMP2521 / COMP9024, i.e., Data Structures and Algorithms.

At the start of this course students should be able to:

- understand fundamental data structures.
- produce correct programs in C/C++, i.e., compilation, running, testing, etc.
- produce readable code with clear documentation.
- appreciate use of abstraction in computing.

Teaching Strategies

There are three primary modes of learning for this course: lectures, written exercises, programming assignments.

Lectures/Exercises

Each week there will be three hours of lectures during which concept/theory, steps, examples and case-studies will be presented. You will get maximum benefits from lectures if you participate in the regular discussions during the lectures.

There will be no tutorials or lab classes in this course, but tutorial-type, written exercises will be made available. You are expected to work through these yourself. Note: none of these are assessable, but we are assuming that you will be interested enough in the topics to actually do them without the need for assessment-based incentives. It is definitely in your own best interest to keep up-to-date with these exercises.

Given the absence of tutorials and labs, you **must** make use of (a) the discussions during the lectures, (b) consultations, (c) Web forums, in order to get your questions answered.

Assignments

There are two programming assignments for this course. They contribute 50% of the overall mark for this course, and are tentatively scheduled as follows:

#	Description	Due	Marks
1	Programming assignment 1 (fundamental)	End of week 5	15%

2	Programming assignment 2 (compression and search)	End of week 10	35%

Assignments will be completed *individually* ; this means that you should do them *yourself* without assistance from others, except for asking advice from the Lecturer or Tutor. As noted above, assignments are the primary vehicle for learning the material in this course. If you don't do them, or simply copy and submit someone else's work, you have wasted a valuable learning opportunity.

Assignments are to be submitted via "give" before midnight on the due date. Assessment of assignments will be primarily based on how accurately they satisfy the requirements; this means that most of the mark will be based on automatic marking. However, we will also manually examine submitted assignments to determine (a) whether they are written with good style, (b) how closely they satisfied the requirements, if they fail the auto-marking. Late submissions will have marks deducted from the maximum achievable mark at the rate of roughly 1% of the total mark per hour that they are late (i.e., 24% per day), and no submissions will be accepted after 3 days late.

Teaching Rationale

The learning foci in this course are primarily lectures (introduction, illustration, discussion), written exercises (theoretical knowledge) and programming assignments (practical knowledge). The course will have an emphasis on the fundamental techniques and their advantages / disadvantages in the context of real applications. Students will learn the main contents of the course through lectures. Some written exercises are available to assist students to obtain in-depth understanding of course materials.

Finally, the primary learning focus in this course are the programming assignments, which have been designed to be challenging and relevant (i.e., aim to strengthen their understanding and develop practical skills on Web data compression and search).

There are no allocated tutorials and no laboratory classes in this course, because we recognise that students already have significant demands on their time, and formal classes are not necessarily the best way to force people to learn. However, don't make the mistake of thinking that "no classes" means that you can get by with "no work". We will be providing some exercises and we *expect you will do* in your own time.

Naturally, sometimes people will "get stuck" on understanding various aspects of the material, so we have Forums where you can ask questions that can be precisely explained, and where we can share these Q&As with other students. We also run face-to-face consultations each week where you can come if you have a problem that can't be handled easily using the Forums.

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 (mailto:stureps.ug@cse.unsw.edu.au) .

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

There will be two places where your learning in this course will be assessed: assignments and the final exam.

Much as we dislike conflating the *learning* aspect of assignments with their *assessment* aspect, there will be marks for the assignment work. We would rather use the assignments entirely as learning vehicles and have no assessment associated with them, but I suspect that wouldn't result in a satisfactory outcome (i.e., nobody would do them).

In addition to the programming assignments, the final exam will also be a major assessment in this course and aims to test what you learned about data compression and search during the course of the semester. To pass this course, you are required to have satisfactory performance on both the programming assignments and the final exam. In order words, you will not be able to pass this subject even you achieve full marks out of all the programming assignments (i.e., a total mark of 50% for the course overall). Similarly, if you do not attempt any programming assignments, you will not pass this subject even you obtain full marks in your final exam. The following formula describes precisely how the final mark will be computed and how the hurdle (satisfactory performance on both the exam and assignments) will be enforced:

```
ass1        = mark for assignment 1      (out of 30)
ass2        = mark for assignment 2      (out of 70)
exam        = mark for final exam        (out of 100)

FinalMark  = 2 * (ass1 + ass2) * exam / (ass1 + ass2 + exam)

FinalGrade =
FL, if FinalMark < 50/100
PS, if 50/100 ≤ FinalMark < 65/100
CR, if 65/100 ≤ FinalMark < 75/100
DN, if 75/100 ≤ FinalMark < 85/100
HD, if FinalMark ≥ 85/100
```

Students are eligible for a Supplementary Exam if and only if:

- they cannot attend the final exam due to illness or misadventure
- the final mark is in the range 45 ≤ FinalMark < 50 (in this case, FinalMark is limited to 50)

A Supplementary Exam will not be awarded for any other reason.

Course Schedule

The order that topics are covered in this course is probably not the most "natural order" for presenting them. The material is presented in lectures in an order that ensures that you are best prepared for the assignments.

The following is an **approximate** guide to the sequence of topics in this course. It is subject to change as the semester progresses.

Week	Lectures	Assignments
1	Introduction, basic information theory	
2	Arithmetic coding, adaptive coding, dictionary coding	
3	Adaptive Huffman, LZW; Overview of BWT	Assignment 1 Released
4	Pattern matching and regular expression	
5	FM index, backward search, suffix array	Assignment 1 Due
6	Suffix array O(n), compressed BWT	Assignment 2 Released
7	XML overview; XML compression	
8	Graph compression	
9	Assignment and Exercise Q&A	
10	Distributed Web query processing	Assignment 2 Due
11	Trends and optional topics	
12	Course Revision	

Resources for Students

There will be no textbook used in this course. Lecture slides and supplementary readings will be provided and used.

You may find the readings below useful as reference materials:

- Managing Gigabytes: Compressing and Indexing. Documents and Images, Second Edition. Ian H. Witten, Alistair Moffat, Timothy C. Bell, Morgan Kaufmann, 1999. (recommended reference, available at the university bookstore)
- Search Engines: Information Retrieval in Practice. W. Bruce Croft, Donald Metzler, and Trevor Strohman, Pearson Education, 2009.

- http://www.data-compression.info (http://www.data-compression.info/) contains lots of valuable resources on data compression (especially links to readings and useful advice), despite the website's pink color!
- Data on the Web: from relations to semistructured data and XML. Serge Abiteboul, Peter Buneman, Dan Suciu. Morgan Kaufmann, 2000.

You will also find your previous textbooks on data structures and/or algorithms useful, in case you need to refer to the fundamentals of data structures and algorithms for text processing.

Last but not least, we will heavily refer to the W3C specs on:

- XML (http://www.w3.org/TR/xml/)
- XML Information Set (http://www.w3.org/TR/xml-infoset/)
- DOM (http://www.w3.org/TR/REC-DOM-Level-1/level-one-core.html)
- XPath (http://www.w3.org/TR/xpath)
- XPath Data Model (XDM) (http://www.w3.org/TR/xpath-datamodel/)
- XQuery (http://www.w3.org/TR/xquery/)

Course Evaluation and Development

This course is evaluated each session using MyExperience.

The MyExperience evaluation from the last time I taught this course showed that students were satisfied with all aspects of the course. Thus we intend to maintain the same style and structure for the up-coming offering, with minor changes to prepare for the UNSW trimester system in 2019. Please note that your feedback is important and will be considered seriously. Student feedback via MyExperience will enable improvements to future offerings of this subject.

Students are also encouraged to provide informal feedback during the semester, and let the lecturer know of any problems, as soon as they arise. Suggestions will be listened to very openly, positively, constructively and thankfully, and every reasonable effort will be made to address them as soon as possible.

Resource created Thursday 19 July 2018, 04:00:14 PM (2018-07-19T16:00:14.881902), last modified Wednesday 17 October 2018, 09:55:58 PM (2018-10-17T21:55:58.439436).