Course Outline Public

Edit Resource (/COMP6741/19T3/resources/32438/edit)

(/COMP6741/18s2/timetable) Contents

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

Course Details

Course Code	COMP6741
Course Title	Parameterized and Exact Computation
Units of Credit	6
Course Website	http://cse.unsw.edu.au/~cs6741 (http://cse.unsw.edu.au/~cs6741)
Handbook Entry	http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6741.html (http://www.handbook.unsw.edu.au/undergraduate/courses/current/COMP6741.html)
Lecturer in charge	Serge Gaspers

Course Summary

The course focuses on algorithms for exactly solving NP-hard computational problems. Since no polynomial time algorithm is known for any of these problems, the running time of the algorithms will have a super-polynomial dependence on the input size or some other parameter of the input.

Taking into account that the difficulty of solving an instance may not only depend on its size n, a parameter k is associated with each instance. The parameterized complexity framework aims at designing fixed-parameter

algorithms whose running times are f(k)*poly(n) for a computable function f. This gives efficient algorithms for small values of the parameter obtained via techniques such as branching, colour coding, and kernelization (preprocessing). We will also see problems that are not fixed-parameter tractable and not kernelizable to polynomial kernels subject to complexity-theoretic assumptions. A large focus is on the design and the worst-case analysis of such algorithms, especially of branching algorithms via the measure & conquer method. We will also see lower bounds for algorithms and how to rule out certain running times assuming the (Strong) Exponential Time Hypothesis.

Course Timetable

The course timetable is available here (https://webcms3.cse.unsw.edu.au/COMP6741/19T3/timetable) .

Consultations: by request; preferred time Wed 4pm.

Course Aims

NP-hard problems are often at the core of the most challenging, rewarding, and lucrative computational problems in all areas of science and technology. This course will outline principled ways to approach these problems and will give students a better understanding of when and why NP-hard computational problem can be solved with reasonable resources.

Student Learning Outcomes

After completing the course, students are able to

- design and analyze non-trivial exponential time algorithms for NP-hard problems using a variety of algorithmic methods
- prove that certain problems cannot be solved in subexponential time or faster than a specific exponential time bound assuming the (Strong) Exponential Time Hypothesis
- design parameterized algorithms for NP-hard problems using a variety of algorithmic methods
- prove that certain parameterizations of problems are not fixed-parameter tractable unless FPT = W[1]

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

Graduate Capability
scholarship: capable of independent and collaborative enquiry
scholarship: rigorous in their analysis, critique, and reflection
scholarship: able to apply their knowledge and skills to solving problems
scholarship: capable of effective communication
scholarship: information literate
scholarship: digitally literate
leadership: enterprising, innovative and creative
leadership: capable of initiating as well as embracing change
leadership: collaborative team workers
professionalism: capable of independent, self-directed practice
professionalism: capable of lifelong learning
global citizens: capable of applying their discipline in local, national and international contexts

Assumed Knowledge

Before commencing this course, students should have basic knowledge in algorithms and complexity, acquired in the course COMP3121 (http://cgi.cse.unsw.edu.au/%7Ecs3121/) .

Teaching Strategies

- Lectures ... introduce concepts, show examples, contain a tutorial part that allows students to solve problems in groups and discover concepts on their own
- Assignments ... allow students to solve significant problems
- Exams ... open-book midterm and final exam

Teaching Rationale

This course is taught by interleaving lectures with exercises, enhances problem-solving capabilities and fortifies the understanding of newly learned concepts.

The assignments further enhance problem-solving at the students' own pace.

The exams confront students with additional problems and test their understanding of concepts and problem-solving capabilities.

Assessment

Assignments. There are six assignments. Assignments 1, 2, 3, and 6 are individual assignments. Assignments 4 and 5 are group assignments: the lecturer-in-charge partitions the students into groups of about 3 students based on their performance in Assignments 1 and 2 (a group consists of students who performed similarly in the assignments). Each student receives one mark for each individual assignment and two marks for each group assignment. The assignment mark of a student is the average of the top seven of these marks.

Late Assignment Submission Policy. Submitting x days after the deadline, with $x > 0$, reduces the grade by $20 \cdot x$ per cent.

Exams. The midterm and the final exam are written open-book exams.

Assignments	25%
Midterm	25%
Final exam	50%

To pass the course, the final mark, calculated as outlined above, needs to be at least 50. Students in their final year of studies achieving a final score of 47-49 can take a supplementary oral exam in which they have to achieve a mark of at least 50 to pass with a final mark of 50.

Special Consideration

If your work in this course is affected by unforeseen adverse circumstances, you should apply for Special Consideration through MyUNSW, including documentation on how your work has been affected. If your request is reasonable and your work has clearly been impacted, then

- for an assignment, you may be granted an extension, and
- for an exam, you may be offered a supplementary exam.

Note the use of the word "may". None of the above is guaranteed. It depends on you making a convincing case that the circumstances have clearly impacted your ability to work.

If you are registered with Disability Services, please forward your documentation to the lecturer-in-charge within the first two weeks of the semester.

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)

Course Schedule

Indicative schedule of lecture topics

Week	Lectures
1	Introduction; review of NP-completeness
2	Kernelization; approximation algorithms; (integer) linear programming
3	Kernelization; basics of Parameterized Complexity
4	Parameterized intractability
5	Branching algorithms
6	Treewidth
7	-
8	Randomized algorithms; iterative compression
9	Exponential time hypothesis
10	Heuristics and local search; review
11	-

Assignment schedule

Assignment	announced on	submit by	feedback on
1	19 Sep	24 Sep	4 Oct
2	26 Sep	1 Oct	11 Oct
3	3 Oct	~~8 Oct~~ 9 Oct	18 Oct
4 (group)	10 Oct	~~18 Oct~~ 25 Oct	~~28 Oct~~ 04 Nov
5 (group)	24 Oct	12 Nov	22 Nov
6	14 Nov	19 Nov	29 Nov

Mid-session quiz

The mid-session quiz is on Wednesday, 23 Oct 2019, at 2pm (Week 6).

Resources for Students

Recommended readings

- [CFK+15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh. Parameterized Algorithms (http://parameterized-algorithms.mimuw.edu.pl) . Springer, 2015.
- [FK10] Fedor V. Fomin and Dieter Kratsch. Exact Exponential Algorithms (http://www.ii.uib.no/~fomin/BookEA/) . Springer, 2010.

Other useful resources

- [FLSZ19] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Kernelization: Theory of Parameterized Preprocessing (https://kernelization.ii.uib.no) . Cambridge University Press, 2019.
- [DF13] Rodney G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity (https://ebookcentral-proquest-com.wwwproxy1.library.unsw.edu.au/lib/unsw/detail.action?docID=1591891) . Springer, 2013.
- [FG06] Jörg Flum and Martin Grohe. Parameterized Complexity Theory (https://ap01-a.alma.exlibrisgroup.com/view/action/uresolver.do?operation=resolveService&package_service_id=24287812180001731&institutionId=1731&customerId=172 . Springer-Verlag, 2006.
- [G10] Serge Gaspers. Exponential Time Algorithms: Structures, Measures, and Bounds (http://www.cse.unsw.edu.au/~sergeg/SergeBookETA2010_screen.pdf) . VDM Verlag Dr. Mueller, 2010.
- [N06] Rolf Niedermeier. Invitation to Fixed-Parameter Algorithms (https://dx.doi.org/10.1093/acprof:oso/9780198566076.001.0001) . Oxford University Press, 2006.
- FPT Wiki (http://fpt.wikidot.com)

Course Evaluation and Development

This course is evaluated each session using the myExperience system.

In the previous offerings of this course, no substantial deficiencies were notes. Students did however make helpful comments on how to improve the course.

Based on their comments, improvements have been made to the course handouts for both black-and-white and color printing, the structure of lectures, a timeline has been set for feedback on assignments, and the number of assignments has increased while their length has decreased.

Resource created 8 months ago (Friday 06 September 2019, 11:14:55 AM), last modified 5 months ago (Thursday 14 November 2019, 03:49:11 PM).

Comments

(/COMP6741/19T3/forums/search?forum_choice=resource/32438)

(/COMP6741/19T3/forums/resource/32438)

There are no comments yet.