# Course Outline

## Contents


- Course Details
- Course Summary
- Course Timetable
- Course Aims
- Student Learning Outcomes
- Assumed Knowledge
- Teaching Strategies
- Teaching Rationale
- Assessment
- Academic Honesty and Plagiarism
- 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) |

## 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.

The first part presents algorithmic techniques to solve NP-hard problems provably faster than brute-force in the worst case, such as branching algorithms, dynamic programming across subsets, inclusion-exclusion, local search, and measure & conquer. We will also see lower bounds for algorithms and how to rule out certain running times assuming the (Strong) Exponential Time Hypothesis.

Whereas the first part presents "default" algorithms that one would use without knowing much about the instances one is about to solve, the second part acknowledges that the complexity of an instance does not only depend on its size n. A parameter k is associated with each instance and 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, iterative compression, 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.

## Course Timetable

The course timetable is available here (/COMP6741/15s2/timetable) .

## 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]
- rule out polynomial kernels for certain parameterizations of problems

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 the way it is because 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

There are three assignments. Assignments 1 and 2 are group assignments (2-3 students). Assignment 3 is an individual assignment. Each student receives one mark for Assignment 1, two marks for Assignment 2, and one mark for Assignment 3. The Assignments mark of a student is the average of the top three of these marks.

The midterm and the final exam are open-book.

| | |
|---|---|
| Assignments | 25% |
| Midterm | 25% |
| Final exam | 50% |

## Academic Honesty and Plagiarism

**Plagiarism** is defined as (https://student.unsw.edu.au/what-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:

- Learning Centre: Plagiarism and Academic Integrity (http://www.lc.unsw.edu.au/plagiarism/index.html)
- MyUNSW: Plagiarism (https://student.unsw.edu.au/plagiarism) and Academic Misconduct (https://www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf)
- CSE: Addendum to UNSW Plagiarism Guidelines (http://www.cse.unsw.edu.au/~chak/plagiarism/plagiarism.html)
- CSE: Yellow Form (http://www.cse.unsw.edu.au/people/studentoffice/policies/yellowform.html) (whose terms you have agreed to)

Make sure that you read and understand these. Ignorance is not accepted as an excuse for plagiarism.

## Course Schedule

Schedule of lecture topics

| Week | Lectures |
|---|---|
| 1 | Introduction; Dynamic programming |
| 2 | Branching algorithms |
| 3 | Branching algorithms; Inclusion-Exclusion |
| 4 | Inclusion-Exclusion; Basics of Parameterized Complexity |
| 5 | Kernelization |
| 6 | Parameterized Branching Algorithms |
| 7 | Midterm; Parameterized Intractability: the W-hierarchy |
| 8 | Parameter Treewidth |
| 9 | Iterative Compression |
| 10 | Randomized methods and color-coding; Kernel lower bounds |
| 11 | Kernel lower bounds; Exponential time hypothesis |
| 12 | Exponential time hypothesis; Review of covered topics |
| 13 | no lecture |

Assignment schedule

| Assignment | announced on | submit by | feedback on |
|---|---|---|---|
| 1 | 05 August 2015 | 25 August 2015 | 02 September 2015 |
| 2 | 26 August 2015 | 06 October 2015 | 13 October 2015 |
| 3 | 07 October 2015 | 20 October 2015 | 04 November 2015 |

The midterm is on 09 September 2015 (Week 7).

## 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. Springer, 2015.
- [DF13] Rodney G. Downey and Michael R. Fellows. Fundamentals of Parameterized Complexity. Springer, 2013.
- [FK10] Fedor V. Fomin and Dieter Kratsch. Exact Exponential Algorithms. Springer, 2010.

Other useful resources

- [FG06] Jörg Flum and Martin Grohe. Parameterized Complexity Theory. Springer-Verlag, 2006.
- [G10] Serge Gaspers. Exponential Time Algorithms: Structures, Measures, and Bounds. VDM Verlag Dr. Mueller, 2010.
- [N06] Rolf Niedermeier. Invitation to Fixed-Parameter Algorithms. Oxford University Press, 2006.
- FPT Wiki (http://fpt.wikidot.com)

Links (https://www.cse.unsw.edu.au/~cs6741/books.html) to full texts of eBooks that are available through the UNSW Library.

## Course Evaluation and Development

This course is evaluated each session using the CATEI 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, and a timeline has been set for feedback on assignments.

Resource created 6 years ago (Thursday 23 July 2015, 07:43:32 PM), last modified 6 years ago (Wednesday 29 July 2015, 09:19:32 AM).

Comments (/COMP6741/15s2/forums/search?forum_choice=resource/1144) (/COMP6741/15s2/forums/resource/1144)

There are no comments yet.