CUHK-Shenzhen
Course Outline

1. Course Identity

A. Course as listed in CUHK-Shenzhen

The information in this block should be exactly as approved by CUHK Senate. In case there are any differences, please explain in the table below.

Course code	CSC6011
Course title (English)	Theory of Computation
Course title (Chinese)	计算理论
Units	3
Description (English)	This course introduces the theory of computational complexity and standard complexity classes, a cornerstone of theoretical computer science (TCS). Students will learn how to classify the computational problems according to how difficult they are to solve computationally. Key topics include automata and language theory, Turing machines, the P versus NP problem, time/space complexity, and randomized computation. Other topics may include polynomial time hierarchy, Boolean circuits, the complexity of counting problems, and so on. In addition, some research-oriented topics will be covered, such as hardness of approximation, interactive proofs, communication complexity, average-case complexity, and computational learning theory.
Description (Chinese)	本课程介绍计算复杂性和标准复杂性类的基础理论。该课程是理论计算机科学的基石。学生将学习如何根据计算问题的难易程度对计算问题进行分类。 重要主题包括自动机与语言理论、图灵机、P 与 NP 问题、时间/空间复杂性、随机计算。其它主题可能包括多项式时间层次结构、布尔电路、计数问题的复杂性等。 此外，还将涵盖一些面向研究的主题，比如近似难度、交互式证明、通信复杂度、平均案例复杂性和计算学习理论。

B. Corresponding course in CUHK

Please give details of the *closest* corresponding course in CUHK (as approved by CUHK Senate and listed in course list). If the course in Shenzhen maps to more than one course in CUHK, please make multiple copies of the block below.

Course code	CSCI3360
Course title (English)	Introduction to Computational Complexity
Course title (Chinese)	計算複雜度導論
Units	3

Description (English)	This course offers a deep dive into the field of computational complexity, a cornerstone of theoretical computer science (TCS) that studies the inherent difficulty of computational problems and classifies them according to their computational resources, such as time and space. The course is designed to elucidate the pivotal role of computational complexity in shaping the landscape of modern computing, touching on its interplay with algorithm design, cryptography, network, and the burgeoning field of quantum computing. By exploring these interdisciplinary connections, students will gain a holistic view of how complexity theory informs and is informed by other scientific and technological domains, thereby enhancing their ability to innovate and solve complex problems. Topics include: - Resources for computation (time, space, nondeterminism, randomness) and their associated complexity classes - Relationships among resources (P vs NP and more), Reductions, and completeness - Space complexity: PSPACE, L, NL - Randomized computation: RP, BPP - Alternation: the polynomial hierarchy, time-space trade-offs for SAT - Basic circuit complexity (P/poly, NC, and more) - Interactive proofs (AM, MA, IP) - Probabilistically checkable proofs (PCP) and non-approximability - Counting: #P, Toda's Theorem, approximate counting - Average-case complexity and cryptography: TFNP, Kolmogorov complexity, one-way functions, Factoring, Discrete log, Diffie-Hellman - Quantum complexity classes: BQP, QMA, PostBQP, quantum auxiliary advice, hidden subgroup problems, (local) Hamiltonian problems
Description (Chinese)	本科深入探討計算複雜性領域，這是理論計算機科學 (TCS) 的基石，研究計算問題的固有難度，並根據計算資源（例如時間和空間）對它們進行分類。本科旨在闡明計算複雜性在塑造現代計算格局中的關鍵作用，涉及其與演算法設計、密碼學、網路和新興的量子計算領域的相互作用。透過探索這些跨學科聯繫，學生將全面了解複雜性理論如何影響其他科學技術領域以及如何被其他科學技術領域所影響，從而提高他們的創新和解決複雜問題的能力。 主題包括： - 計算資源（時間、空間、不確定性、隨機性）及其相關的複雜性類別 - 資源之間的關係（P 與 NP 等）、縮減與完整性 - 空間複雜度：PSPACE、L、NL - 隨機計算：RP、BPP - 交替：多項式層次結構、SAT 的時空權衡

	- 基本電路複雜度（P/poly、NC 等） - 互動式校樣（AM、MA、IP） - 機率可檢查證明（PCP）和不可近似性 - 計數：#P、戶田定理、近似計數 - 平均情況複雜性與密碼學：TFNP、Kolmogorov 複雜性、單向函數、因式分解、離散對數、Diffie-Hellman - 量子複雜度類別：BQP、QMA、PostBQP、量子輔助建議、隱藏子群問題、（局部）哈密頓問題

2. Prerequisites / Co-requisites

Please state prerequisites and co-requisites, in terms of courses in CUHK-Shenzhen* or any other requirements (e.g., having taken certain subjects in high school).

(* Because course codes may not yet be stable, please provide both course code and course tile.)

A. Prerequisites

No formal prerequisites. Students should be comfortable with mathematical reasoning, proofs, and basic discrete mathematics.

B. Co-requisites

3. Learning Outcomes

By the end of this course, students will be able to:
(1) Formalize computational problems using standard models of computation, analyze the capabilities and limitations of these models, and prove results regarding decidability, reducibility, and computational complexity.
(2) Classify problems into major complexity classes, construct rigorous reductions and impossibility arguments.
(3) Apply theoretical-computer-science concepts to the analysis of algorithms and data-science problems.

4. Course syllabus

Automata and Language Theory (3 weeks):
- Regular languages, finite automata (DFA and NFA).
- Context-free languages, pushdown automata.

Computability Theory (3 weeks):
- Turing machines, the Church-Turing thesis.

- Decidability, diagonalization.
- Reduction.

Complexity Theory (6 weeks):
- Time complexity, P and NP, NP-completeness, Cook-Levin theorem.
- Space complexity, PSPACE, L, NL, Savitch's theorem.
- Randomized computation, RP, BPP.
- Interactive proofs, IP=PSPACE.
- Approximation algorithms, hardness of approximation, PCP theorem.

Other Topics (2 weeks):
- Communication complexity.
- Transformers and EXPSPACE-Completeness

5. Assessment Scheme

Component/ method	% weight
Homework	30%
Midterm Exam	30%
Final Exam	40%

6. Grade descriptor

Grade	Description
A	EXCELLENT – exceptionally good performance and far exceeding expectation in all or most of the course learning outcomes; demonstration of superior understanding of the subject matter, the ability to analyze problems and apply extensive knowledge, and skillful use of concepts and materials to derive proper solutions.
A-	GOOD – good performance in all course learning outcomes and exceeding expectation in some of them; demonstration of good understanding of the subject matter and the ability to use proper concepts and materials to solve most of the problems encountered.
B	SATISFACTORY – satisfactory performance that meets expectations in most course learning outcomes; demonstration of sound understanding of the subject matter and the ability to solve standard problems.
C	FAIR – adequate performance that meets basic expectations in the essential course learning outcomes; demonstration of adequate understanding of the subject matter and the ability to solve simple problems.
D	MARGINAL – performance that only just meets the minimum expectations in the essential course learning outcomes; demonstration of limited but sufficient understanding of the subject matter and the ability to solve basic problems with guidance.
F	FAILURE – performance does not meet the expectation in the essential course learning outcomes; demonstration of serious deficiencies and the need to retake the course.

7. Reading

A. Required

None

B. Recommended

1. "Introduction to the Theory of Computation", by Michael Sipser. ISBN: 978-1-133-18779-0. Publisher: Cengage Learning
2. "Introduction to Automata Theory, Languages, and Computation", by John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. ISBN: 978-0-321-45536-9. Publisher: Pearson.
3. "Computational Complexity: A Modern Approach", by Sanjeev Arora and Boaz Barak. ISBN: 978-0-521-42426-4. Publisher: Cambridge University Press.
4. "Communication Complexity", by Eyal Kushilevitz and Noam Nisan. ISBN: 978-0-511-57494-8. Publisher: Cambridge University Press.

8. Course components

Activity	Hours/week
Lectures	3 h/w
Office hour by instructor	1 h/w
Office hour by TA	1 h/w

9. Indicative teaching plan

Week	Content/ topic/ activity
1	Introduction. Finite automata (DFA and NFA). Regular languages.
2	Nonregular languages, pumping lemma.
3	Context-free languages, pushdown automata.
4	Turing machines, the Church-Turing thesis.
5	Decidability, diagonalization.
6	Reduction. Time complexity.
7	P and NP, NP-completeness.
8	Cook-Levin theorem. [Midterm Exam]
9	Space complexity, PSPACE, L, NL, Savitch's theorem.
10	[ICLR'26 paper]: Transformers & EXPSPACE-completeness.
11	Randomized computation, RP, BPP.
12	Interactive proofs, IP=PSPACE.
13	Approximation algorithms, hardness of approximation, PCP theorem.

14	Communication complexity.

10. Any other information

Students must complete midterm and final exams independently, without using AI tools. Exams are open-book: students can bring text materials but not digital devices.
AI tools can be used in homework assignments. However, students should write the homework solutions by themselves and indicate how they use AI. Direct submission of AI outputs is not allowed.

11. Version date

Version number	2
As of (date)	2026-09-07