Introduction to the Theory of Computation

Department: Fudan International Summer Session 2025

<table>
<tr><td>Course Code</td><td colspan="7">TBA</td></tr>
<tr><td>Course Title</td><td colspan="7">Introduction to the Theory of Computation</td></tr>
<tr><td>Credit</td><td>2</td><td>Experiment (including Computer) Credit</td><td>0</td><td>Practice Credit</td><td>0</td><td>Aesthetic Education Credit</td><td>0</td></tr>
<tr><td>Credit Hours Per Week</td><td>9 credit hours per week, 36+3 tutorial hours in total (one credit hour is 45 minutes)</td><td>Education on The Hard-Working Spirit Credit Hours</td><td>0</td><td>Language of Instruction</td><td>English</td><td>Honors Course</td><td>□Yes
☑No</td></tr>
<tr><td rowspan="2">Course Type</td><td colspan="3" rowspan="2">□Core General Education Course
□Specific General Education Course
□Basic Course in General Discipline
☑Others</td><td colspan="4">2+X Major ：
□Professional Core Course
□Professional Advanced Course</td></tr>
<tr><td colspan="4">Non 2+X Major ：
□Professional Compulsory Course
□Professional Elective Course</td></tr>
<tr><td>Course Objectives</td><td colspan="7">（Including value, knowledge and ability objectives）
Computation is everywhere. Often time, people overlook the theory behind it.

This course will give you a beginner-friendly introduction to the Theory of Computation. The Theory of Computation seeks to categorize computational problems based on their inherent difficulty, measured by the resources (primarily time and space) needed to solve them. It also aims to explore the relationships between different problems, such as determining whether problem X is not harder than problem Y. This course will help you gain a rigorous understanding of computation, including its definition, possibilities, and limitations.

Topics include finite automaton and regular language, Turing machine and its variants,</td></tr>
</table>

	computability, and complexity theory. This course is exclusively for the Fudan International Summer School program.
Course Description	By the end of this course, you will have a clue about the answers to the following questions: - What is computation? How to define computation models rigorously? - What functions are computable? - What can be computed efficiently?

Course Requirements:

You need a background in linear algebra, calculus, and some basic programming experience.

Teaching Methods:

The instructor will give in-class lectures, give self-contained proofs for the key results, and handwrite all the board notes.

The instructor will ask questions, provide in-class exercise, to ensure students have understood the materials.

Course Director's Academic Background:

Yuan Li is an assistant professor at Fudan University. Prior to this, he was a software engineer at Google from 2017 to 2022. He received his BSc in Computer Science from Fudan University in 2011 and his PhD in Computer Science from the University of Chicago in 2017. His research interests include computational complexity and coding theory. He has published papers in conferences and journals such as IEEE Transactions on Information Theory, FOCS, SIAM Journal on Computing, Information and Computation, and DCC (Designs, Codes and Cryptography).

Instructor's Academic Background:

Same as above.

Members of Teaching Team

Name	Gender	Professional Title	Department	Responsibility
Yuan Li	male	assistant professor	computer science	teaching

Course Schedule (Please supply the details about each lesson)**:**

Lecture 1. Big O notation, alphabet and languages

Lecture 2. Finite automaton

Lecture 3. Regular language

Lecture 4. Turing machine and its variants

Lecture 5. Universal Turing machine, computability

Lecture 6. Turing halting problem, undecidable problems

Lecture 7. Reductions, Church – Turing thesis

Lecture 8. Complexity class P, NP

Lecture 9. Cook-Levin theorem

Lecture 10. Turing reduction and Cook reduction

Lecture 11. More NP complete problems

Lecture 12. Final exam

The design of class discussion or exercise, practice, experience and so on:

In each class, the instructor will give in-class exercises to make sure you grasp the course materials.
You will gain a hand-on experience programming a Turing machine using a Turing machine simulator.
Throughout the course, questions are encouraged and can be asked via email or in the course WeChat group, where the instructor or teaching assistant will respond.

If you need a TA, please indicate the assignment of assistant:

Grade homework assignments and the final exam.
Respond to questions via email or in the WeChat group.

Grading & Evaluation (Provide a final grade that reflects the formative evaluation process)**:**

Homework: 5 assignments, each worth 6 points
Exams: 50 points. We will have a final exam in the course. The final exam will be held in class.
In-class participation: 10 points
Attendance: 10 points

Usage of Textbook： ☑Yes(complete textbook information form below) ☐No

Textbook Information (No more than two textbooks)**:**

Title	Author	ISBN	Publishing Time	Publisher	Type I	Type II
Introduction to the Theory of Computation	Michael Sipser	113318779X	June 27, 2012	Cengage Learning	□Self-compiled Textbook (Published) ☑Non-mainland Textbook □Other Textbook (Published)	□National Planning Textbook □Provincial and Ministerial Planning Textbook □School Level Planning Textbook □Others

Teaching References (Including author, title, publisher, publishing time,ISBN)**:**
Sanjeev Arora, Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, April 20, 2009, 0521424267.

Table column size can be adjusted according to the content.