eecs2001

Introduction to the Theory of Computation

Lecture 1

Ruth Urner

January 9, 2023

Organization of the class

- Instructor: Ruth Urner (ruth@eecs.yorku.ca)
- Emails: start your subject line with [eecs2001].

- Office Hours: I will stay available for questions on zoom after each class and tutorial.

- Website: available on eClass.

Lectures

Lecture times Monday, Wednesday, 1:00-2:20pm.

Tutorial times Friday, 1:00-2:20pm.

Meeting place Zoom; meeting ID available on eclass.

First lecture Today :) January 9.

Last lecture Wedneday, April 5.

Last tutorial Monday, April 10
(make-up Friday for Good Friday on April 7).

Reading week February 18-24 (week 7).

Evaluation (to be confirmed before January 23)

4 or 5 Assignments (30%)

- Spaced throughout the term
- Exact dates will be announced soon

In-person Midterm exam (30%)

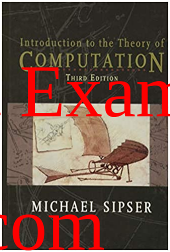
- Expect in the week after reading week
- Date to be announced once we have a room

In-person Final exam (40%)

- Date to be scheduled by the university
- Examination period: April 12-27

Introduction to the Theory of Computation (ITC)

by Michael Sipser (third edition).



- Available on Amazon.
- Available from York U bookstore.

Please get the text book. While I will aim to provide slides with a self-contained presentation of the material it can be useful to be able to consult an additional resource. I will aim to keep presentation and notation consistent with this textbook. I would discourage consulting additional resources, since encountering deviating notation or terminology can be confusing when first learning a subject. Rather, take time and patience to learn the material from the text book, lectures, slides, class videos and always feel free to ask questions when something is unclear!

- I will hold lectures at the announced times on zoom.

- Preliminary versions of the lecture slides will be posted to eclass before the lectures. I will update these during the 24 hours after the lectures.

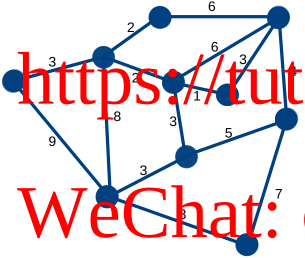
- Tutorial meetings will be in person in DB 0016 and also on the same zoom meeting. Practice questions for each tutorial will be posted/announced during the week and solutions will be presented during the tutorial.

Please always feel free to ask questions if anything is unclear!

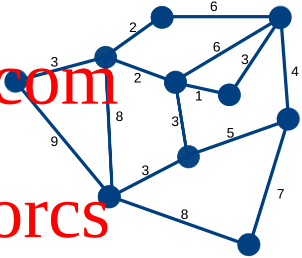
Theory of Computation–Motivation

Reading: ITC Section 0.1

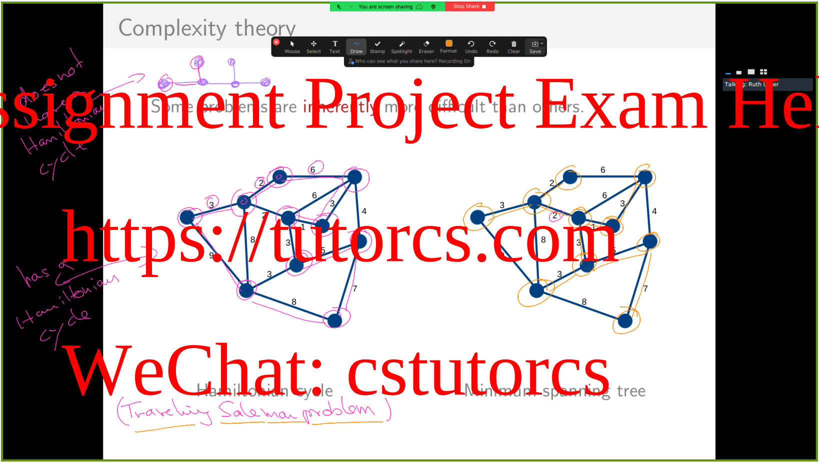
Some problems are inherently more difficult than others.



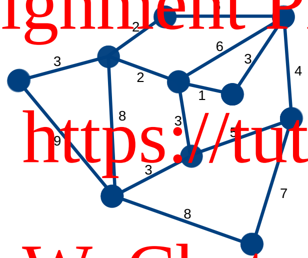
Hamiltonian cycle



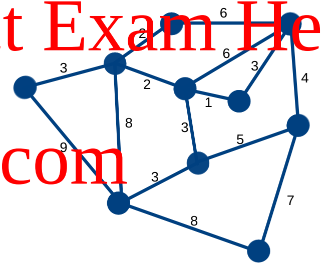
Minimum spanning tree


You are screen sharing
Stop Share
Complexity theory
Mouse Select Text Draw Stamp Spotlight Eraser Format Undo Redo Clear Save
Who can see what you share here? Recording On
Some problems are inherently more difficult than others.
does not have a Hamiltonian cycle
has a Hamiltonian cycle
Hamiltonian cycle
(Traveling Salesman problem)
Minimum spanning tree

Some problems are inherently more difficult than others. They have different computational complexity.



Hamiltonian cycle
Very difficult (NP-hard) problem



Minimum spanning tree
Easy problem,
can be solved in O(m log(n)) steps

→ In this course, we will learn how to formalize computational problems and analyze and compare their computational difficulty (complexity).

Why is it important to understand the computational complexity of a problem?

- If you have a solution (algorithm) for a problem, you may want to know if your solution is optimal, or whether it could be improved.
- Many safety critical applications rely on computational hardness for security and privacy.
  - In cryptography we want a proof that some encryption scheme is safe!

Some problems are so difficult that no algorithm can solve them...

Question:
Can one write a program $\mathrm{HALT}(P, i)$ that, when it gets the code of another program P checks whether program P will halt or loop forever when run on input i?

Answer:
No. One can prove that such a program can not exist. The halting problem is uncomputable.

Complexity theory and computability theory require precise definitions of problems and computers.

Automata theory provides models of computation and the theory of formal languages provides ways to formalize what computational problems are.

→ We will use finite automata as a starting point for understanding how computation can be studied and understood in a mathematically sound way.

→ We start by reviewing some basic mathematical concepts, notation and terminology.

Basic mathematical notation and terminology

Reading: ITC Section 0.2

Set-theory is the foundation of mathematics.

We will start with the set of natural numbers as a basic given set to start with:

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

Further, we will assume the existence of the empty set: $\emptyset$

Sets

Set-theory is the foundation of mathematics.

We will start with the set of natural numbers as a basic given set to start with:

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}$$

Further, we will assume the existence of the empty set: $\emptyset$

$\emptyset$ does not contain any elements

$\emptyset$ is a subset of every other set

A set consists of elements, denoted by the $\in$-relation:

$$a \in A$$

means a is an element of set A.

To state that a is not an element of set A, we use the notation

$$a \notin A.$$

Important properties:

- Sets are not ordered and contain each element only once!
- The empty set has no elements.

Elements of sets

A set consists of elements, denoted by the $\in$-relation:

$$a \in A$$

means a is an element of set A.

To state that a is not an element of set A, we use the notation

$$a \notin A.$$

Important properties:

- Sets are not ordered and contain each element only once!

$\{2,4,6,4\} = \{2,4,6\} = \{4,6,2\} = \{6,4,2,2\}$

Subsets

Using the element relation, we can define what subsets are:

- $A \subseteq B$

- $A \subsetneq B$

- $A = B$

Subsets

Using the element relation, we can define what subsets are:

- $A \subseteq B$ "A is a subset of B"

  If $c \in A$ then $c \in B$.

  $\{1,2,4\} \subseteq \mathbb{N}$, $\{1,2,4\} \subseteq \{1,2,4\}$

- $A \subsetneq B$ "A is a proper subset of B"

  $A \subseteq B$ and there is a $b \in B$ with $b \notin A$.

  $\{1,2,4\} \subsetneq \mathbb{N}$, $\{n \in \mathbb{N} \mid n \text{ even}\} \subsetneq \mathbb{N}$

- $A = B$ "A and B are equal as sets"

  $A \subseteq B$ and $B \subseteq A$

  $\{1,2,4,2\} = \{4,2,1\}$

How to define sets

Here are some tools for defining sets:

1. Make a list of the elements
   - Set of all students in this class
   - Set of odd natural numbers smaller than 10: $\{1, 3, 5, 7, 9\}$

   Problem: this technique fails for large or infinite sets

2. Identify by a common characteristic
   - Odd natural numbers $\{n \in \mathbb{N} \mid n \text{ is not divisible by } 2\}$

   Problem: sometimes we don't know a precise defining characteristic

3. Inductive definition
   - We'll see how to do this later.

Warning:
When we define sets, we always need to specify from which universe (that is a possibly much larger ground set, for example the natural numbers) the elements of our set should be taken from!

Example:

- Odd natural numbers $\{n \in \mathbb{N} \mid n \text{ is not divisible by } 2\}$
- Interval on the real line $\{x \in \mathbb{R} \mid 2 \leq x \leq 4\}$

Otherwise we can fall into Russell's paradox...!

Consider the following definition of a set:

$$R = \{r \mid r \notin r\}$$

That is, the set R contains all those sets that do not contain themselves as an element.

Question: Is R an element of R?

Now, $R \in R$ implies that $R \notin R$, and vice versa ($R \notin R$ implies that $R \in R$) – a contradiction.

Russel's paradox

Consider the following definition of a set

$$\boxed{R = \{r \mid r \notin r\}}$$

That is, the set R contains all those sets that do not contain themselves as an element.

Question: Is R an element of R?

Now, $R \in R$ implies that $R \notin R$, and vice versa ($R \notin R$ implies that $R \in R$)– a contradiction.

Set operations

By default, we will for now assume that all our sets are subsets of the natural numbers. That is we assume a universe $U = \mathbb{N}$. Then we can define:

- The set-difference $A \setminus B$ of two sets A and B

- The intersection $A \cap B$ of two sets A and B

- The union $A \cup B$ of two sets A and B

Set operations

By default we will for now assume that all our sets are subsets of the natural numbers. That is we assume a universe $U = \mathbb{N}$. Then we can define:

$A - B$

- The set-difference $\boxed{A \setminus B}$ of two sets A and B

always: $A \setminus B \subseteq A$

$A \setminus B = \{a \in A \mid a \notin B\}$

$A = \{1, 5, 17\}$ … $= \{1, 2, 3\}$

$A \setminus B = \{5, 17\}$