

# **Lecture 1**

Course Overview, Administrative Details, Logic

# Course Overview

# Course Overview

Goal of this course:

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

**Tentative Plan:**

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

## **Tentative Plan:**

- **Logic:** Propositional logic, Predicates, and Quantification, Natural Deduction.

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

## Tentative Plan:

- **Logic:** Propositional logic, Predicates, and Quantification, Natural Deduction.
- **Proof Techniques:** Introduction to Proofs, Induction, Proof by contradiction.

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

## Tentative Plan:

- **Logic:** Propositional logic, Predicates, and Quantification, Natural Deduction.
- **Proof Techniques:** Introduction to Proofs, Induction, Proof by contradiction.
- **Set Theory and Combinatorics:** Naive Set Theory, Counting, Permutations, Combinations, Partitions, Pigeonhole Principle, Inclusion-exclusion, Binomial theorem.



# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

## Tentative Plan:

- **Logic:** Propositional logic, Predicates, and Quantification, Natural Deduction.
- **Proof Techniques:** Introduction to Proofs, Induction, Proof by contradiction.
- **Set Theory and Combinatorics:** Naive Set Theory, Counting, Permutations, Combinations, Partitions, Pigeonhole Principle, Inclusion-exclusion, Binomial theorem.
- **Graph Theory:** Properties of graphs, Graph Matching and Coloring.

# Course Overview

**Goal of this course:** Learn mathematics that is **useful** and **necessary** for Computer Science.

## Tentative Plan:

- **Logic:** Propositional logic, Predicates, and Quantification, Natural Deduction.
- **Proof Techniques:** Introduction to Proofs, Induction, Proof by contradiction.
- **Set Theory and Combinatorics:** Naive Set Theory, Counting, Permutations, Combinations, Partitions, Pigeonhole Principle, Inclusion-exclusion, Binomial theorem.
- **Graph Theory:** Properties of graphs, Graph Matching and Coloring.
- **Models of Computation:** Finite Automaton, Regular Expressions, Pumping Lemma of Regular Languages, Context Free Grammar, Turing Machines, Undecidability.

# Administrative Details

# Administrative Details

Grading:

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major
- 0% - Practice problems whose variants may come in exam



# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major
- 0% - Practice problems whose variants may come in exam

**Note:** *No requizzes will be conducted. So try to not miss more than 1 quiz.*

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major
- 0% - Practice problems whose variants may come in exam

**Note:** *No requizzes will be conducted. So try to not miss more than 1 quiz.*

**Note:** *Grading will be relative. Score at least 30% to pass the course.*

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major
- 0% - Practice problems whose variants may come in exam

**Note:** *No requizzes will be conducted. So try to not miss more than 1 quiz.*

**Note:** *Grading will be relative. Score at least 30% to pass the course.*

**Office Hours:** Mail me to fix an appointment.

# Administrative Details

## Grading:

- 30% - Quizzes (MCQs, T/Fs) (best 3 out of 4 quizzes)
- 30% - Minor
- 40% - Major
- 0% - Practice problems whose variants may come in exam

**Note:** *No requizzes will be conducted. So try to not miss more than 1 quiz.*

**Note:** *Grading will be relative. Score at least 30% to pass the course.*

**Office Hours:** Mail me to fix an appointment.

**Attendance:** As per the institute policy.

# Administrative Details

# Administrative Details

**Tutorials:** Will be held in two batches:

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.



# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.

**Books:**

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.

**Books:**

- Discrete mathematics and its applications by Kenneth H. Rosen. (*For module 1 and 2*)

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.

**Books:**

- Discrete mathematics and its applications by Kenneth H. Rosen. (*For module 1 and 2*)
- A Walk Through Combinatorics by Miklós Bóna. (*For module 3 and 4*)

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.

**Books:**

- Discrete mathematics and its applications by Kenneth H. Rosen. *(For module 1 and 2)*
- A Walk Through Combinatorics by Miklós Bóna. *(For module 3 and 4)*
- Introduction to Theory of Computation by Michael Sipser. *(For module 5)*

# Administrative Details

**Tutorials:** Will be held in two batches:

- Odd Roll No: 3 PM - 4 PM, Wednesday.
- Even Roll No: 4 PM - 5 PM, Wednesday.

**Books:**

- Discrete mathematics and its applications by Kenneth H. Rosen. (*For module 1 and 2*)
- A Walk Through Combinatorics by Miklós Bóna. (*For module 3 and 4*)
- Introduction to Theory of Computation by Michael Sipser. (*For module 5*)

**Course Site:** <http://home.iitj.ac.in/~vimalraj/courses/csl2040.html>

# Glimpses of the Course

# Glimpses of the Course

We'll learn about the following and more in this course:

# Glimpses of the Course

We'll learn about the following and more in this course:

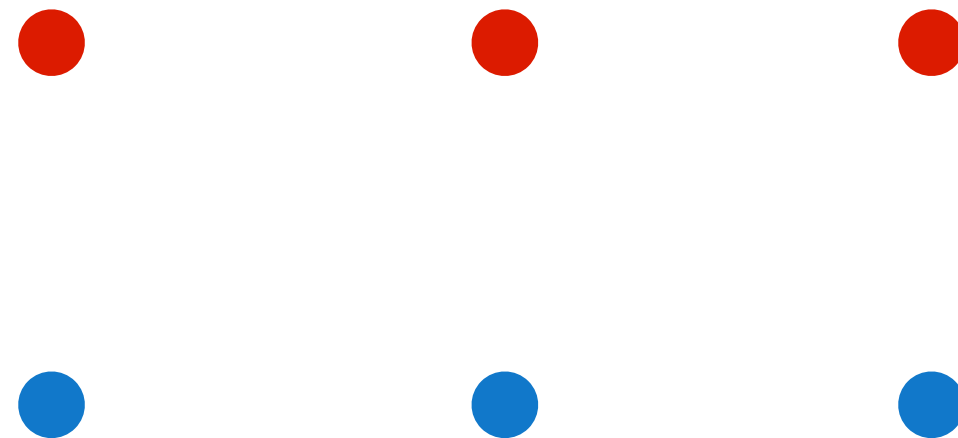
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?



# Glimpses of the Course

We'll learn about the following and more in this course:

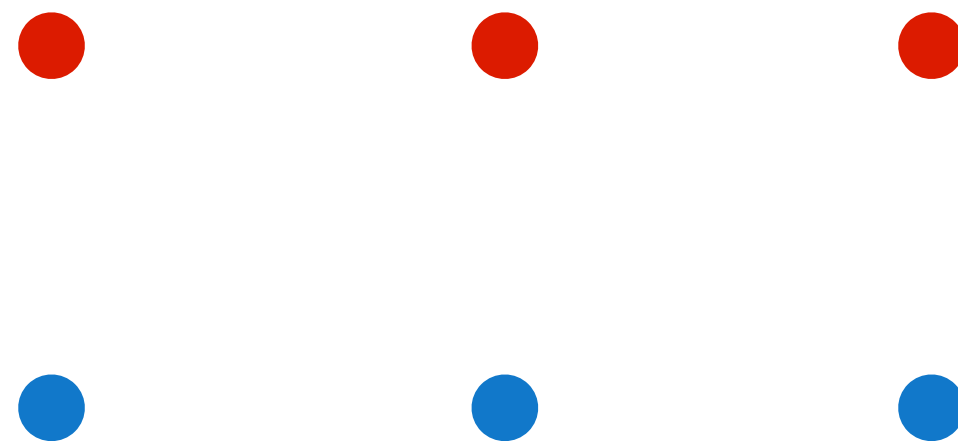
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?



# Glimpses of the Course

We'll learn about the following and more in this course:

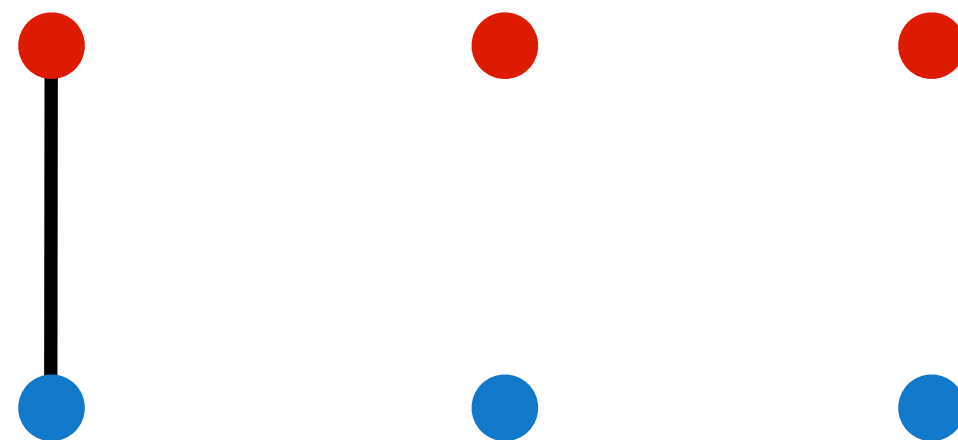
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

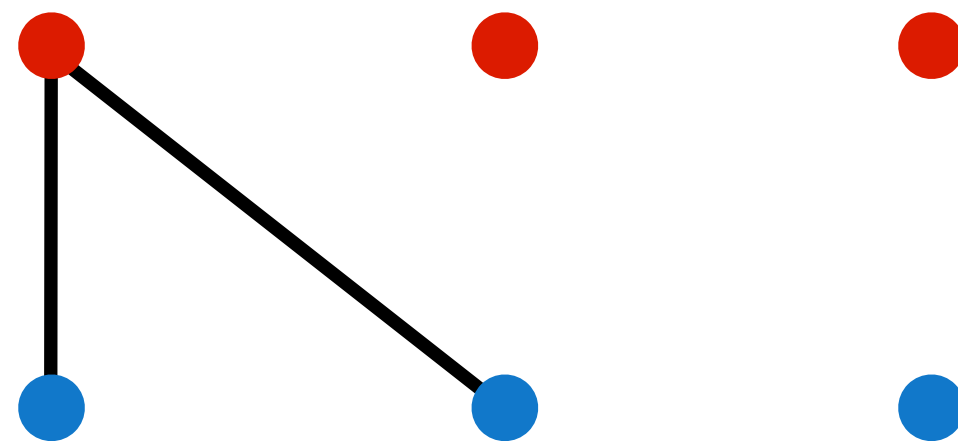
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

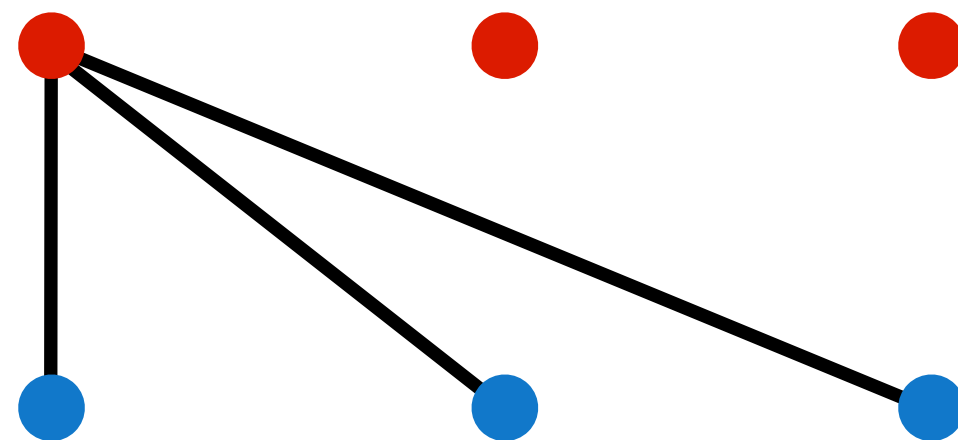
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

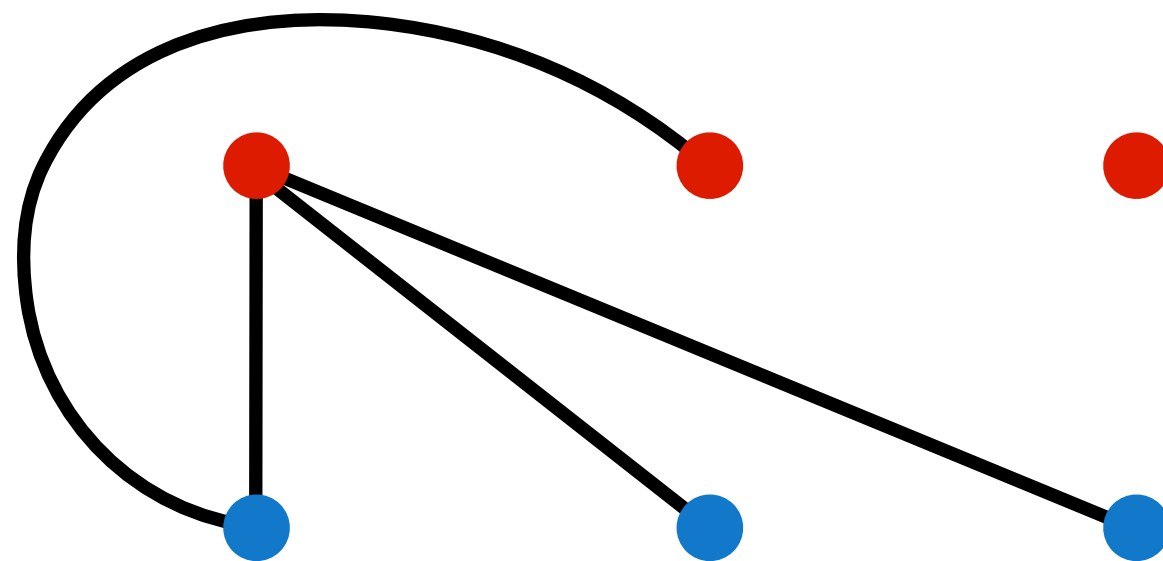
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

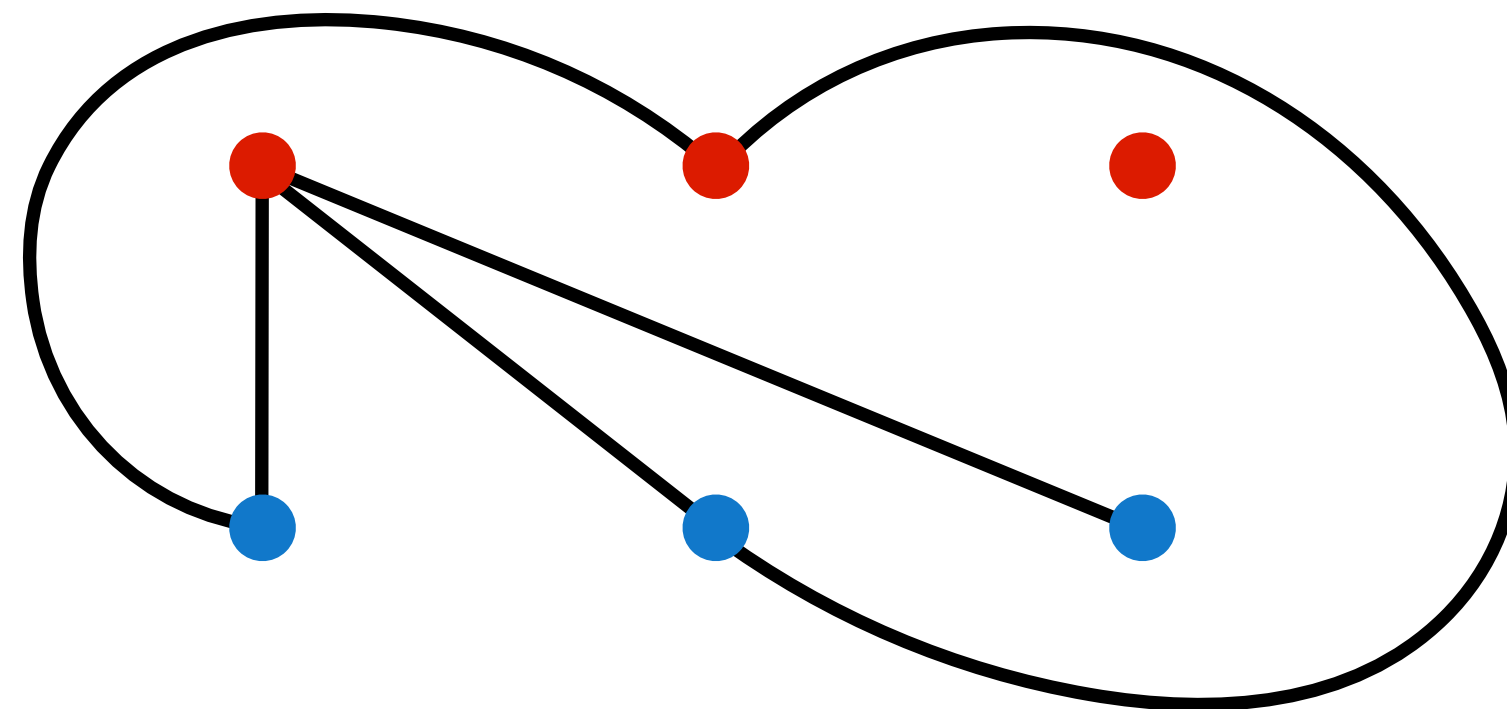
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

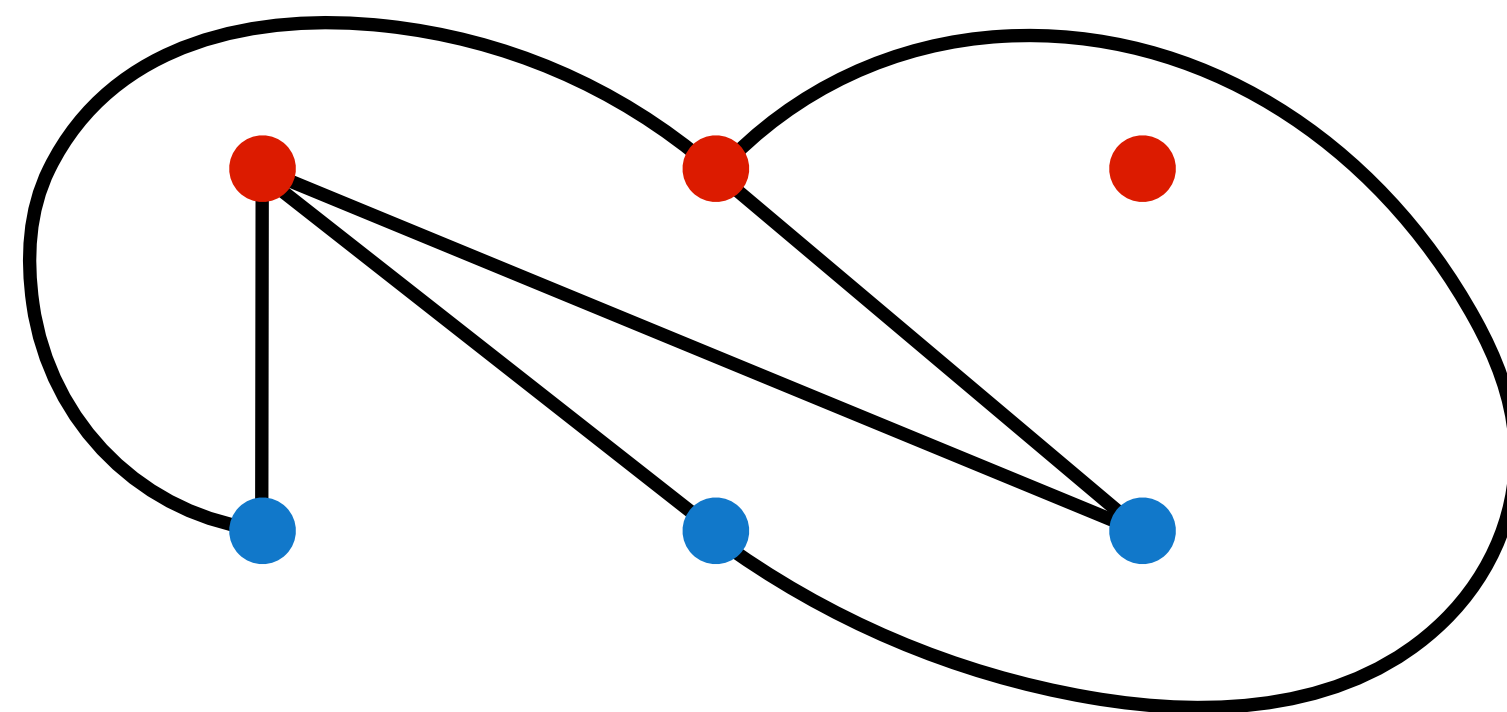
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?

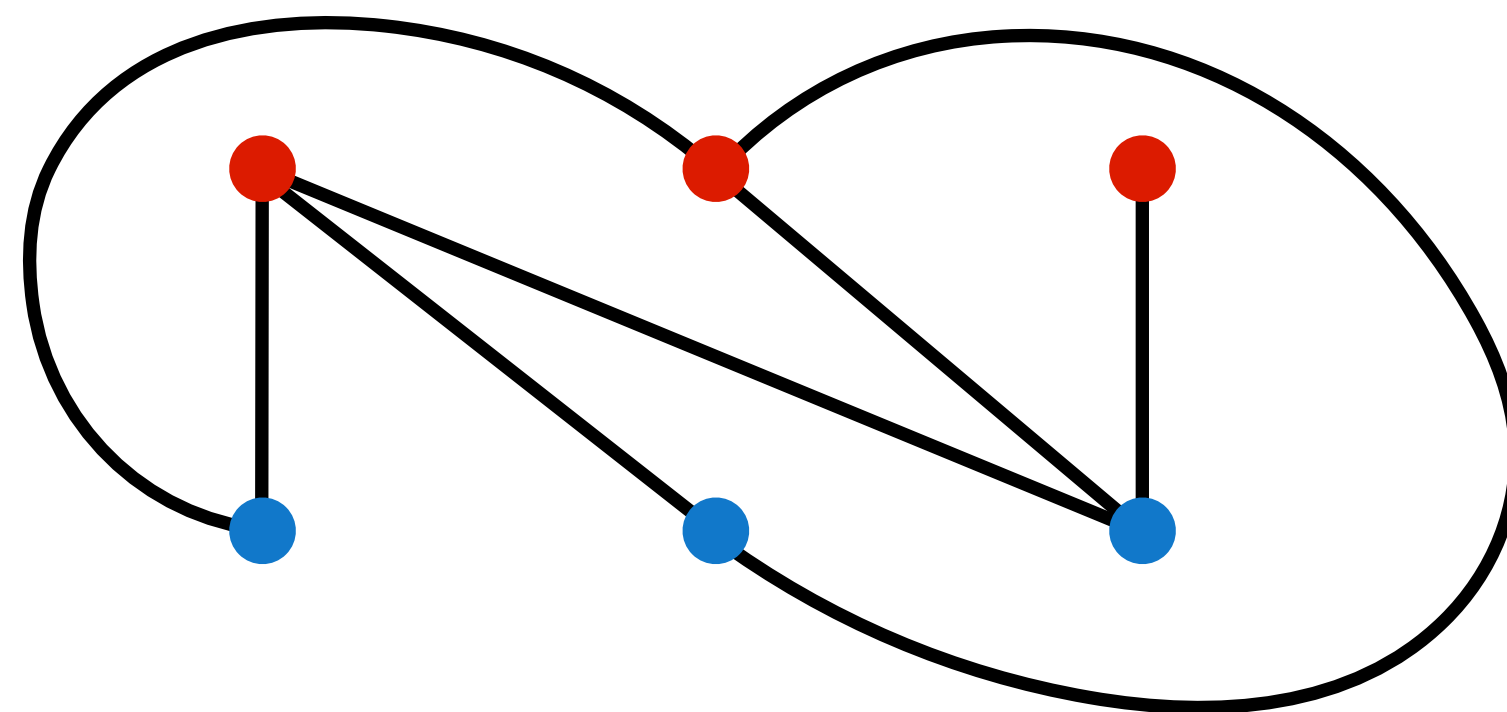




# Glimpses of the Course

We'll learn about the following and more in this course:

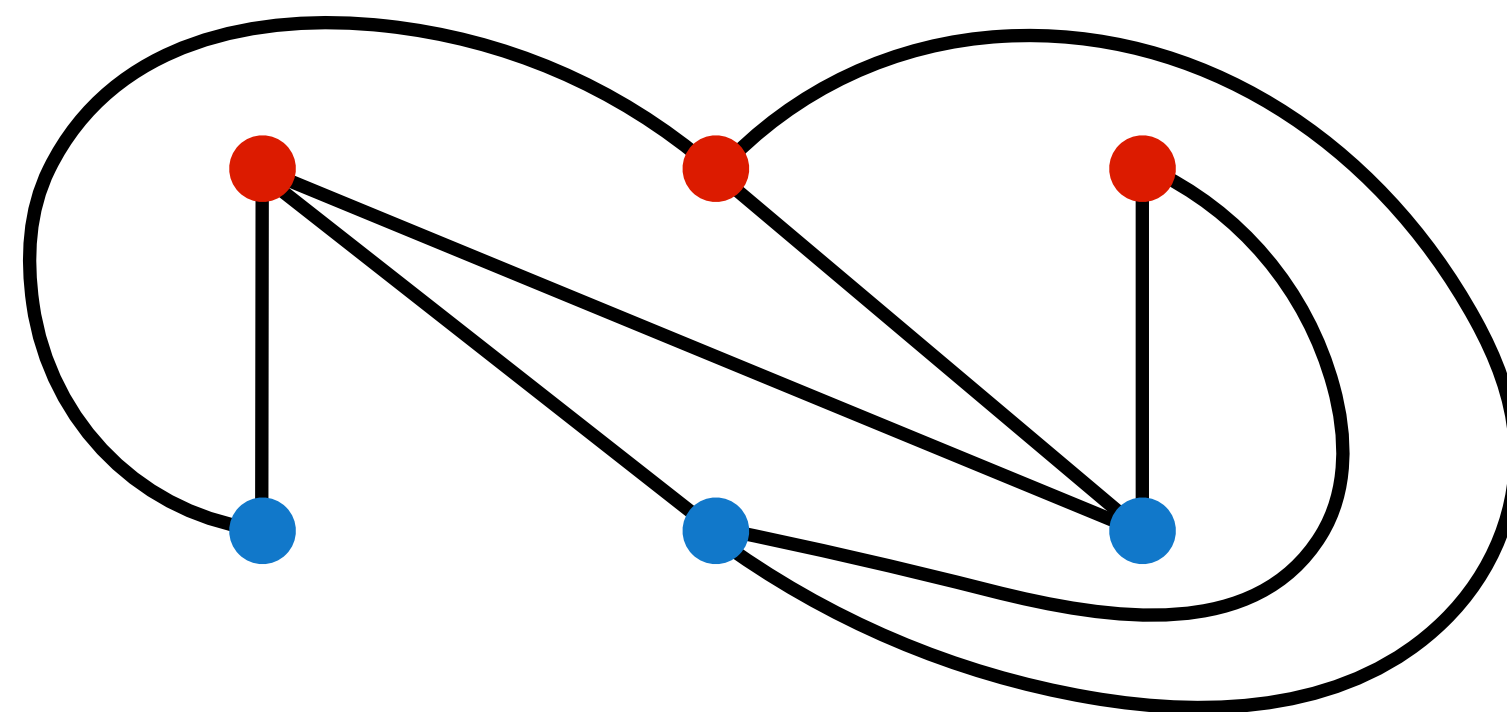
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

We'll learn about the following and more in this course:

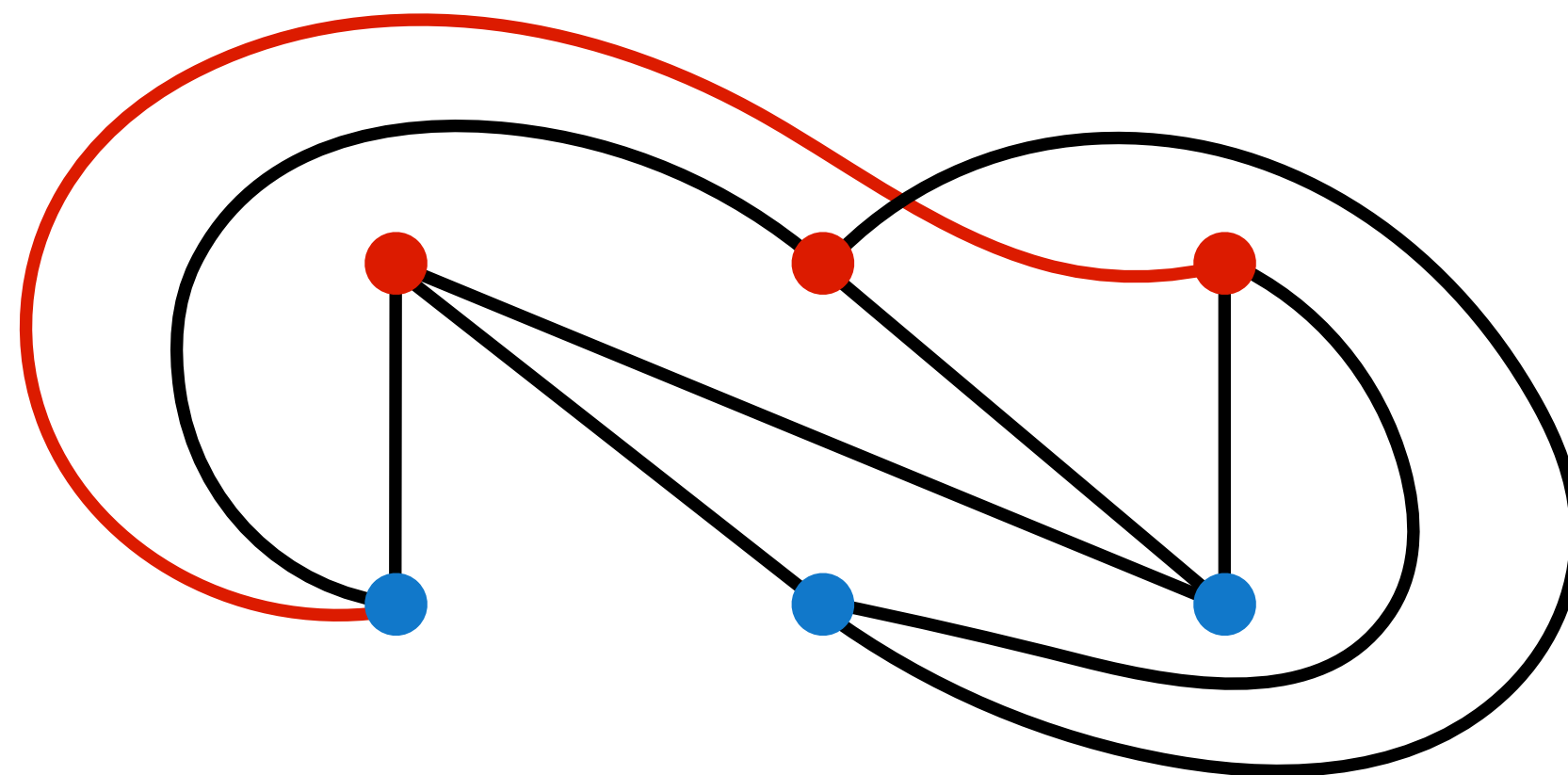
- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



# Glimpses of the Course

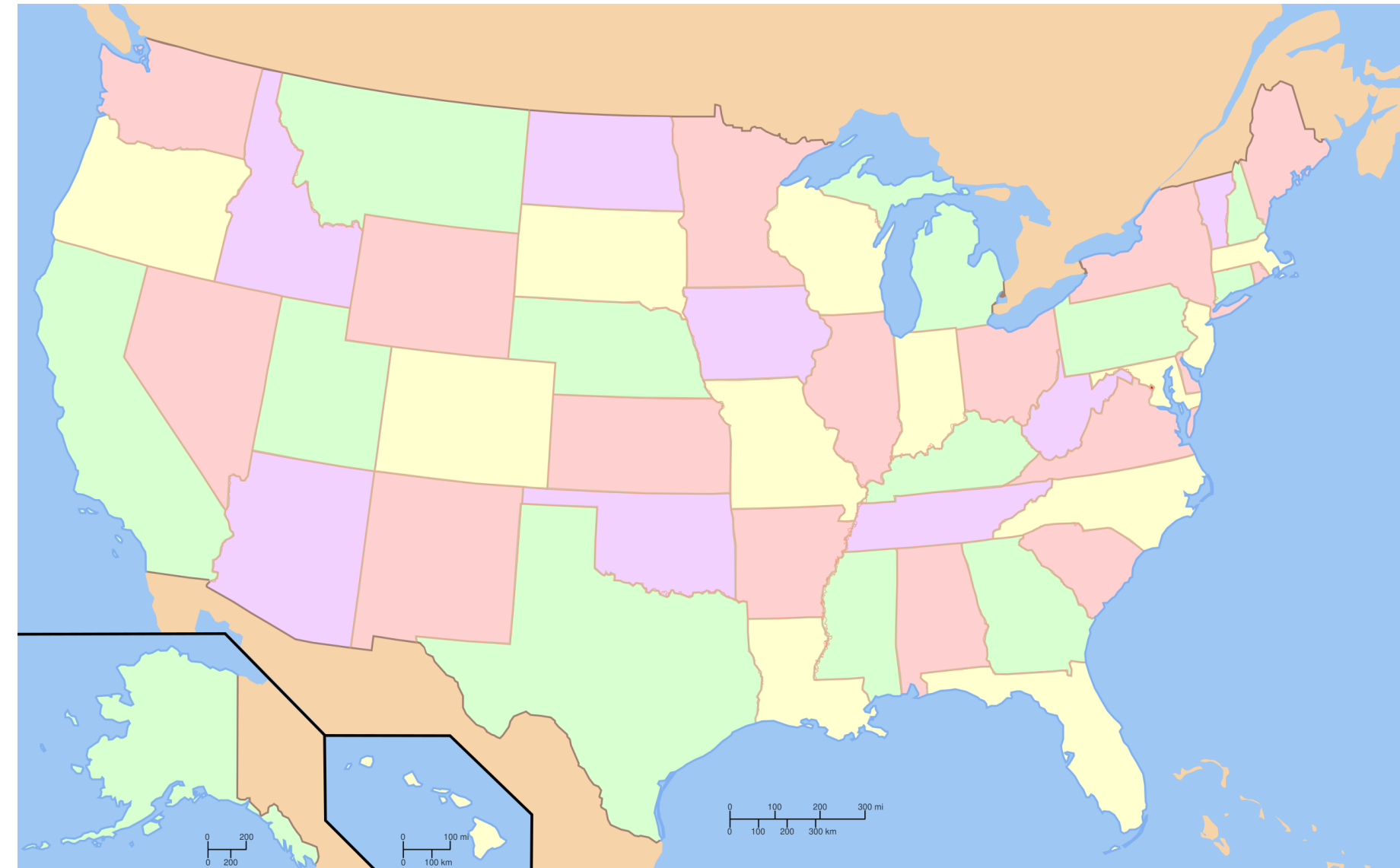
We'll learn about the following and more in this course:

- Are  $\mathbb{Z}$  and  $\mathbb{R}$  equally large?
- Can every ● be connected to every ● with a wire such that no two wires cross each other?



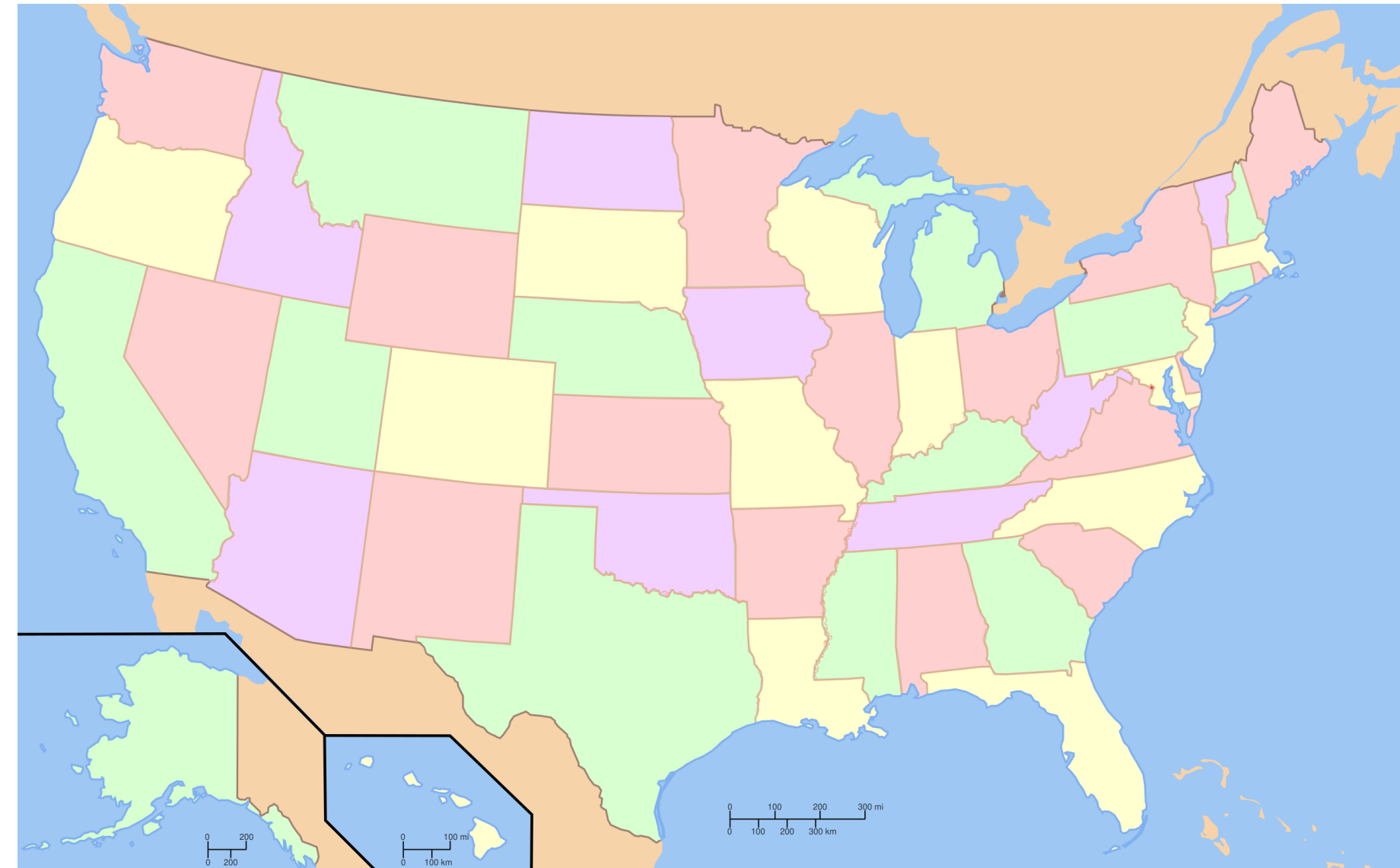
# Glimpses of the Course

# Glimpses of the Course



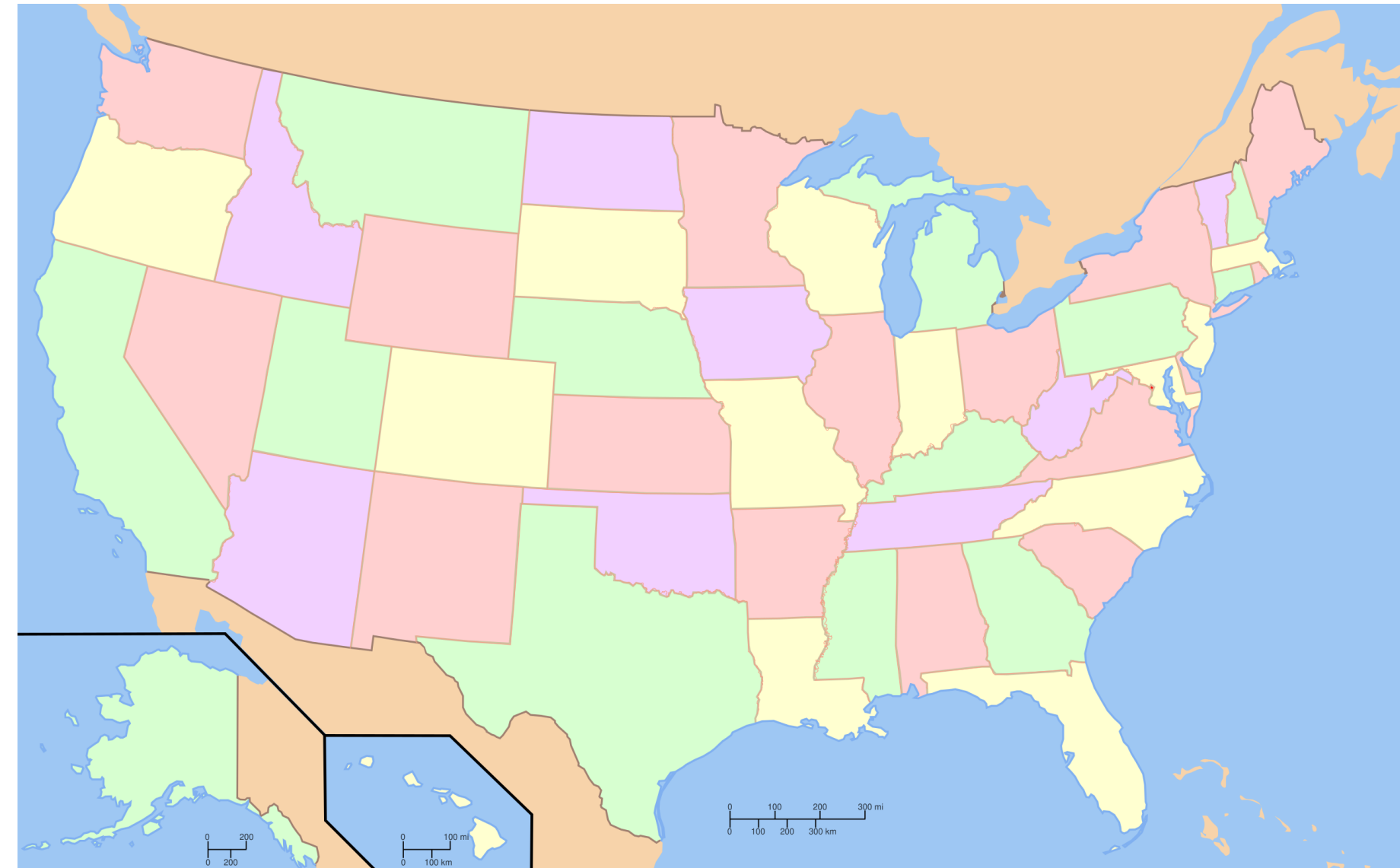
# Glimpses of the Course

- Can we colour a country's map with 4 colours so that no two adjacent states have same colour?



# Glimpses of the Course

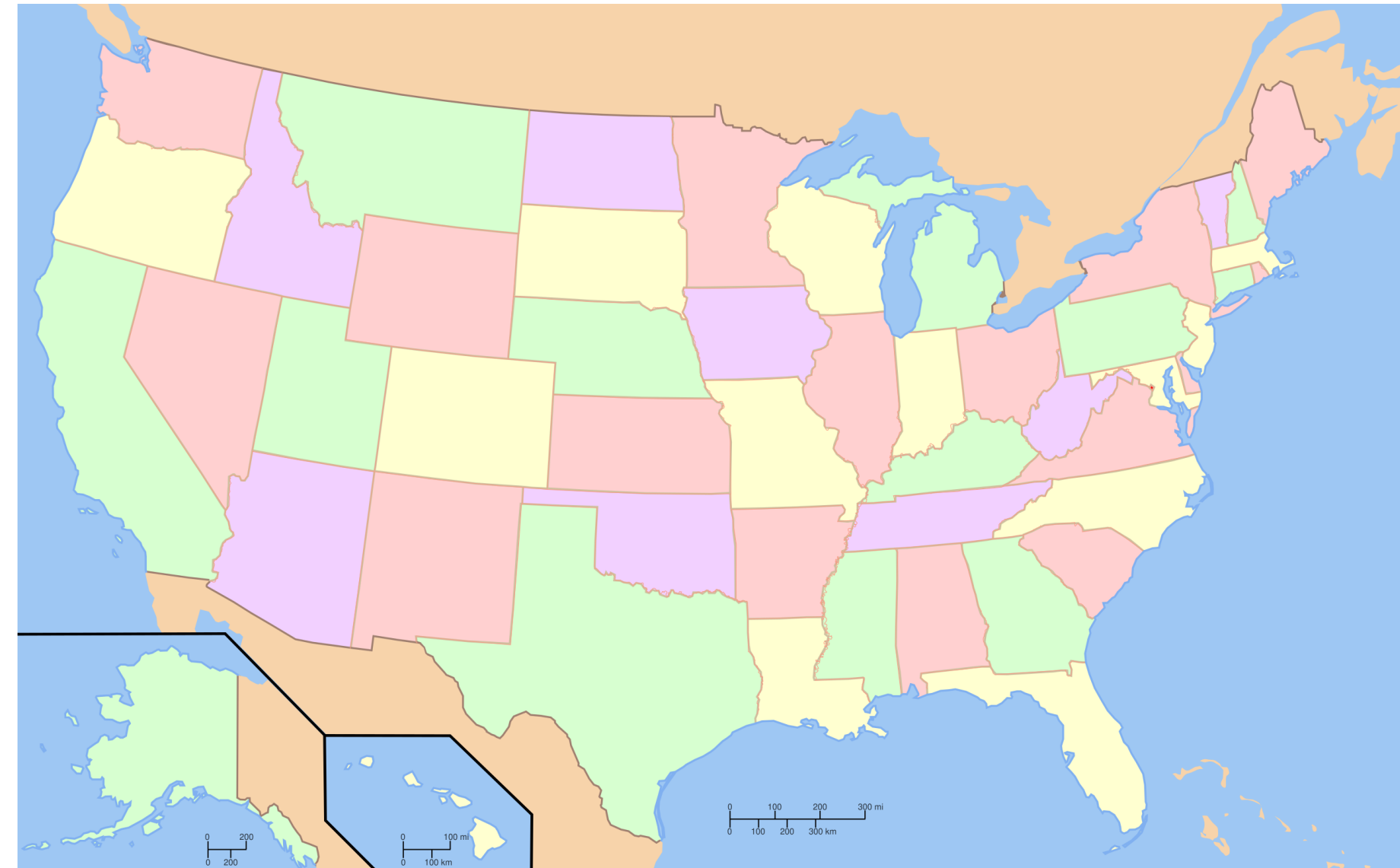
- Can we colour a country's map with 4 colours so that no two adjacent states have same colour?



- Can we write a C program that takes another C program, say **X**, and its input, say **Y**, and

# Glimpses of the Course

- Can we colour a country's map with 4 colours so that no two adjacent states have same colour?



- Can we write a C program that takes another C program, say **X**, and its input, say **Y**, and outputs whether **X** will terminate on **Y**?



# Logic

# Logic

What is logic?

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

## Example 1:

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

#### Assumptions:

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** {

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** { 1. All men are mortal.

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** {  
1. All men are mortal.  
2. Socrates is a man.



# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** { 1. All men are mortal.  
2. Socrates is a man.

**Conclusion:**

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** { 1. All men are mortal.  
2. Socrates is a man.

**Conclusion:** Socrates is mortal.



*valid conclusion*

# Logic

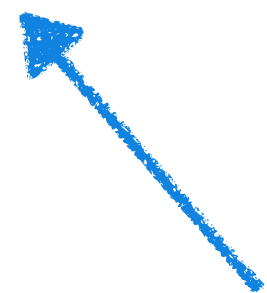
## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** { 1. All men are mortal.  
2. Socrates is a man.

**Conclusion:** Socrates is mortal.



*valid conclusion*

# Logic

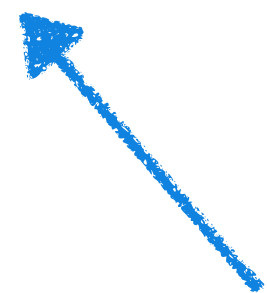
## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.



*valid conclusion*

### Example 2:

# Logic

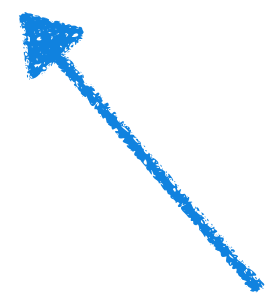
## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** { 1. All men are mortal.  
2. Socrates is a man.

**Conclusion:** Socrates is mortal.



*valid conclusion*

### Example 2:

**Assumptions:**

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:** {  
1. All men are mortal.  
2. Socrates is a man.

**Conclusion:** Socrates is mortal.

  
valid conclusion

### Example 2:

**Assumptions:** {

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.

  
valid conclusion

### Example 2:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All cats are mortal.} \end{array} \right.$



# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.



*valid conclusion*

### Example 2:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All cats are mortal.} \\ 2. \text{ Socrates is mortal.} \end{array} \right.$

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.

  
valid conclusion

### Example 2:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All cats are mortal.} \\ 2. \text{ Socrates is mortal.} \end{array} \right.$

**Conclusion:**

# Logic

## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.

  
valid conclusion

### Example 2:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All cats are mortal.} \\ 2. \text{ Socrates is mortal.} \end{array} \right.$

**Conclusion:** Socrates is a cat.

# Logic

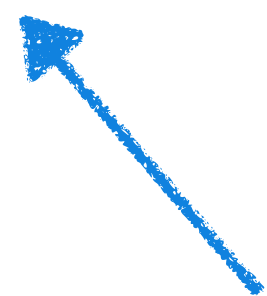
## What is logic?

Logic focuses on reasoning or how to reliably get from **assumptions** to **conclusions**.

### Example 1:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All men are mortal.} \\ 2. \text{ Socrates is a man.} \end{array} \right.$

**Conclusion:** Socrates is mortal.



*valid conclusion*

### Example 2:

**Assumptions:**  $\left\{ \begin{array}{l} 1. \text{ All cats are mortal.} \\ 2. \text{ Socrates is mortal.} \end{array} \right.$

**Conclusion:** Socrates is a cat.



*invalid conclusion*

# Pitfalls of Natural Language in Reasoning

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:**

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** {



# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:**

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:**

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** {  
1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** {



# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** { 1. Nothing is better than eternal happiness.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** { 1. Nothing is better than eternal happiness.  
2. McDonald's burger is better than nothing.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** { 1. Nothing is better than eternal happiness.  
2. McDonald's burger is better than nothing.

**Conclusion:**

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** { 1. Nothing is better than eternal happiness.  
2. McDonald's burger is better than nothing.

**Conclusion:** McDonald's burger is better than eternal happiness.

# Pitfalls of Natural Language in Reasoning

Natural language is sometimes **imprecise**:

**Assumption:** { 1. Only married couples can book a room in hotel Moonlight.  
2. Jim and Pam are married.

**Conclusion:** Jim and Pam can book a room in hotel Moonlight.

**Issue:** May be Jim and Pam are not married to each other?

Natural language is sometimes **ambiguous**:

**Assumptions:** { 1. Nothing is better than eternal happiness.  
2. McDonald's burger is better than nothing.

**Conclusion:** McDonald's burger is better than eternal happiness.

**Issue:** 'Nothing' has different meaning in both assumptions.