

BILL AND EMILY RECORD LECTURE!!!!

**UNTIMED PART OF
FINAL
MONDAY May 10
9:00AM
DEAD CAT
WED May 12 9:00PM**

FINAL IS
MONDAY May 17
8:00PM-10:15PM

**FILL OUT COURSE
EVALS for ALL YOUR
COURSES!!!**

Theory Courses Beyond CMSC 250H

You've Taken CMSC 250H. Now What?

If a, b are courses at UMCP then let $R(a, b)$ mean that b is a good course to take right after a .

You've Taken CMSC 250H. Now What?

If a, b are courses at UMCP then let $R(a, b)$ mean that b is a good course to take right after a .

Let

$$A_0 = \{250H\}.$$

You've Taken CMSC 250H. Now What?

If a, b are courses at UMCP then let $R(a, b)$ mean that b is a good course to take right after a .

Let

$$A_0 = \{250H\}.$$

For all $i \geq 1$ let

$$A_i = A_{i-1} \cup \{b : R(a, b) \wedge a \in A_{i-1}\}$$

You've Taken CMSC 250H. Now What?

If a, b are courses at UMCP then let $R(a, b)$ mean that b is a good course to take right after a .

Let

$$A_0 = \{250H\}.$$

For all $i \geq 1$ let

$$A_i = A_{i-1} \cup \{b : R(a, b) \wedge a \in A_{i-1}\}$$

$$A = \bigcup_{i=1}^{\infty} A_i$$

You've Taken CMSC 250H. Now What?

If a, b are courses at UMCP then let $R(a, b)$ mean that b is a good course to take right after a .

Let

$$A_0 = \{250H\}.$$

For all $i \geq 1$ let

$$A_i = A_{i-1} \cup \{b : R(a, b) \wedge a \in A_{i-1}\}$$

$$A = \bigcup_{i=1}^{\infty} A_i$$

We discuss the courses in A .

CMSC 351: Algorithms

This is a course in Algorithms.

CMSC 351: Algorithms

This is a course in Algorithms.

1. You will learn many algorithms and analyze their run times.

CMSC 351: Algorithms

This is a course in Algorithms.

1. You will learn many algorithms and analyze their run times.
2. How to analyze their run times? With **recurrences**.

CMSC 351: Algorithms

This is a course in Algorithms.

1. You will learn many algorithms and analyze their run times.
2. How to analyze their run times? With **recurrences**.
3. If taken with Clyde Kruskal there will be an NP-completeness project.

CMSC 351: Algorithms

This is a course in Algorithms.

1. You will learn many algorithms and analyze their run times.
2. How to analyze their run times? With **recurrences**.
3. If taken with Clyde Kruskal there will be an NP-completeness project.
4. NP-completeness is about showing that problem likely **do not** have poly time algorithms.

CMSC 351: Algorithms

This is a course in Algorithms.

1. You will learn many algorithms and analyze their run times.
2. How to analyze their run times? With **recurrences**.
3. If taken with Clyde Kruskal there will be an NP-completeness project.
4. NP-completeness is about showing that problem likely **do not** have poly time algorithms.
5. The mathematical maturity you gained in 250H will help you a lot in 351.

CMSC 451: Algorithms

Similar to CMSC 351 with the following difference:

CMSC 451: Algorithms

Similar to CMSC 351 with the following difference:

1. CMSC 351 is about solving **particular problems**.

CMSC 451: Algorithms

Similar to CMSC 351 with the following difference:

1. CMSC 351 is about solving **particular problems**.
2. CMSC 451 is about **algorithmic paradigm**. Example: Dynamic Programming, Greedy algorithms, Divide and Conquer.

CMSC 451: Algorithms

Similar to CMSC 351 with the following difference:

1. CMSC 351 is about solving **particular problems**.
2. CMSC 451 is about **algorithmic paradigm**. Example: Dynamic Programming, Greedy algorithms, Divide and Conquer.
3. There will be even more about NP-completeness

CMSC 451: Algorithms

Similar to CMSC 351 with the following difference:

1. CMSC 351 is about solving **particular problems**.
2. CMSC 451 is about **algorithmic paradigm**. Example: Dynamic Programming, Greedy algorithms, Divide and Conquer.
3. There will be even more about NP-completeness
4. This is taught by a wide variety of people (though not me) all of whom are good to take it with.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

1. Regular Languages- can be recognized with $O(1)$ space and only sees the input once.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

1. Regular Languages- can be recognized with $O(1)$ space and only sees the input once.
2. P, NP, other time classes. Maybe space classes.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

1. Regular Languages- can be recognized with $O(1)$ space and only sees the input once.
2. P, NP, other time classes. Maybe space classes.
3. Decidable, Undecidable, and beyond.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

1. Regular Languages- can be recognized with $O(1)$ space and only sees the input once.
2. P, NP, other time classes. Maybe space classes.
3. Decidable, Undecidable, and beyond.

CMSC 452 is always taught by **William Gasarch**.

CMSC 452: Automata Theory

The basic question in theory is:

Given a Problem, how hard is it to solve?

- ▶ In CMSC 351 and 451 this usually means we want to solve a problem QUICKLY.
- ▶ What about showing that a problem **cannot** be solved quickly?

CMSC 452 is about upper and lower bounds on many different models of computation.

1. Regular Languages- can be recognized with $O(1)$ space and only sees the input once.
2. P, NP, other time classes. Maybe space classes.
3. Decidable, Undecidable, and beyond.

CMSC 452 is always taught by **William Gasarch**.

That means that in a non-pandemic year there will be a lecture where you get to eat muffins.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.
Eve intercepts it.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

This course is cross-listed in 3 depts: CMSC, MATH, ENEE.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

This course is cross-listed in 3 depts: CMSC, MATH, ENEE.

The course is taught by diff people within each dept.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

This course is cross-listed in 3 depts: CMSC, MATH, ENEE.

The course is taught by diff people within each dept.

There is **not** a CMSC-version, MATH-version, ENEE-version.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

This course is cross-listed in 3 depts: CMSC, MATH, ENEE.

The course is taught by diff people within each dept.

There is **not** a CMSC-version, MATH-version, ENEE-version.

There is a **Katz**-version, **Dachman-Soled**-version,

Gasarch-version, **Washington**-version, etc.

CMSC 456: Cryptography

Alice wants to send Bob a message in code.

Eve intercepts it.

Alice and Bob want the code to be unbreakable or hard to break.

1. What if Alice and Bob meet ahead of time?
2. What if Alice and Bob cannot meet?
3. What if Eve has unlimited computational power?
4. What if Eve has only limited computational power?
5. What about Alice and Bob's computational power?
6. Why did my Dentist email me that his files were hacked and some foreign government knows about my weak gums?

This course is cross-listed in 3 depts: CMSC, MATH, ENEE.

The course is taught by diff people within each dept.

There is **not** a CMSC-version, MATH-version, ENEE-version.

There is a **Katz**-version, **Dachman-Soled**-version,

Gasarch-version, **Washington**-version, etc.

Katz and Dachman-Soled teach a more rigorous version than

Gasarch or Washington.

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

1. Can do a few things (not many) faster than classical computers can.

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

1. Can do a few things (not many) faster than classical computers can.
2. Don't really exist yet (this can be debated).

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

1. Can do a few things (not many) faster than classical computers can.
2. Don't really exist yet (this can be debated).
3. Have lead to a better understanding of physics.

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

1. Can do a few things (not many) faster than classical computers can.
2. Don't really exist yet (this can be debated).
3. Have lead to a better understanding of physics.
4. Have lead to a better understanding of computer science.

CMSC 457: Quantum Computing

Cross listed with Physics, though does not require any physics to understand the material.

Quantum computers:

1. Can do a few things (not many) faster than classical computers can.
2. Don't really exist yet (this can be debated).
3. Have lead to a better understanding of physics.
4. Have lead to a better understanding of computer science.

We have three quantum profs, and I they all teach it.

CMSC 752: Ramsey Theory and its “Applications”

If you liked Pigeonhole and grid coloring you will like Ramsey Theory.

CMSC 752: Ramsey Theory and its “Applications”

If you liked Pigeonhole and grid coloring you will like Ramsey Theory.

Sample Theorem For all 2-colorings of \mathbb{N} there exists a, d such that

$a, a + d, \dots, a + 1000d$ are all the same color.

CMSC 752: Ramsey Theory and its “Applications”

If you liked Pigeonhole and grid coloring you will like Ramsey Theory.

Sample Theorem For all 2-colorings of \mathbb{N} there exists a, d such that

$$a, a + d, \dots, a + 1000d \text{ are all the same color.}$$

Only taught by **William Gasarch**. I hope to teach it in Spring 2023 or Spring 2024 but not sure if will be able to.

CMSC 752: Ramsey Theory and its “Applications”

If you liked Pigeonhole and grid coloring you will like Ramsey Theory.

Sample Theorem For all 2-colorings of \mathbb{N} there exists a, d such that

$$a, a + d, \dots, a + 1000d \text{ are all the same color.}$$

Only taught by **William Gasarch**. I hope to teach it in Spring 2023 or Spring 2024 but not sure if will be able to.

Only prereq is math maturity, which you all either have or will have by the next time its taught.

Other Theory Grad Courses

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.
2. Algorithmic Game Theory. Calculating optimal strategies.

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.
2. Algorithmic Game Theory. Calculating optimal strategies.
3. Randomized Algorithms. Algorithms that flip coins and may have a small prob of error.

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.
2. Algorithmic Game Theory. Calculating optimal strategies.
3. Randomized Algorithms. Algorithms that flip coins and may have a small prob of error.
4. Fun with Hardness. Proving that some problems are hard to solve or even approximate.

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.
2. Algorithmic Game Theory. Calculating optimal strategies.
3. Randomized Algorithms. Algorithms that flip coins and may have a small prob of error.
4. Fun with Hardness. Proving that some problems are hard to solve or even approximate.
5. Data Science and Machine Learning Courses are on the border of Theory and AI. Uses probability and statistics.

Other Theory Grad Courses

1. Computational Geometry. Algorithms in geometry. Used in Vision, Graphics.
2. Algorithmic Game Theory. Calculating optimal strategies.
3. Randomized Algorithms. Algorithms that flip coins and may have a small prob of error.
4. Fun with Hardness. Proving that some problems are hard to solve or even approximate.
5. Data Science and Machine Learning Courses are on the border of Theory and AI. Uses probability and statistics.
6. Quantum Computing.