COMP 761: Mathematical Tools for CS

David Rolnick

September 2, 2020

Welcome to COMP-761

This is a whirlwind introduction to important math that turns up everywhere in computer science. We will explore many topics. But the goal is really to learn how to reason mathematically about problems in computer science from the ground up – and how to prove statements formally when you need to.

About the course

About the course

About the course

- Foundational math tools that you will use

About the course

- Foundational math tools that you will use
- Mathematical ways of thinking about problems

About the course

- Foundational math tools that you will use
- Mathematical ways of thinking about problems
- Cool & useful applications to different areas

About the course

- Foundational math tools that you will use
- Mathematical ways of thinking about problems
- Cool & useful applications to different areas
- How to create and write a proof

About the course

- Foundational math tools that you will use
- Mathematical ways of thinking about problems
- Cool & useful applications to different areas
- How to create and write a proof

No background in theoretical math is necessary

Introductions

Outline

Outline

Outline

1. Introduction
2. Proofs and how to think about problems

Outline

Outline

Outline

1. Introduction
2. Proofs and how to think about problems
3. Proof techniques: Induction
4. Proof techniques: Contradiction
5. Other proof techniques (e.g. monovariants and inequalities)

Outline

1. Introduction
2. Proofs and how to think about problems
3. Proof techniques: Induction
4. Proof techniques: Contradiction
5. Other proof techniques (e.g. monovariants and inequalities)
6. Combinatorics (e.g. binomials, combinatorial identities)

Outline

Outline

1. Introduction
2. Proofs and how to think about problems
3. Proof techniques: Induction
4. Proof techniques: Contradiction
5. Other proof techniques (e.g. monovariants and inequalities)
6. Combinatorics (e.g. binomials, combinatorial identities)
7. Graph theory I (e.g. trees, Hamiltonian cycles, colorings)
8. Linear algebra I (e.g. vector spaces, matrix products, determinants, SVD)

Outline

1. Introduction
2. Proofs and how to think about problems
3. Proof techniques: Induction
4. Proof techniques: Contradiction
5. Other proof techniques (e.g. monovariants and inequalities)
6. Combinatorics (e.g. binomials, combinatorial identities)
7. Graph theory I (e.g. trees, Hamiltonian cycles, colorings)
8. Linear algebra I (e.g. vector spaces, matrix products, determinants, SVD)
9. Linear algebra II (e.g. matrix norms, random matrix theory)

Outline

1. Introduction
2. Proofs and how to think about problems
3. Proof techniques: Induction
4. Proof techniques: Contradiction
5. Other proof techniques (e.g. monovariants and inequalities)
6. Combinatorics (e.g. binomials, combinatorial identities)
7. Graph theory I (e.g. trees, Hamiltonian cycles, colorings)
8. Linear algebra I (e.g. vector spaces, matrix products, determinants, SVD)
9. Linear algebra II (e.g. matrix norms, random matrix theory)
10. Graph theory II (e.g. random walks, graph Laplacians, spectral graph theory)

Outline

Outline

11. Calculus I (e.g. limits, derivatives, gradients)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)
16. Linear programs II (e.g. simplex method, ellipsoid method)

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)
16. Linear programs II (e.g. simplex method, ellipsoid method)
17. Semidefinite and quadratic programs

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)
16. Linear programs II (e.g. simplex method, ellipsoid method)
17. Semidefinite and quadratic programs
18. Integer programs

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)
16. Linear programs II (e.g. simplex method, ellipsoid method)
17. Semidefinite and quadratic programs
18. Integer programs
19. Spanning trees, shortest paths

Outline

11. Calculus I (e.g. limits, derivatives, gradients)
12. Calculus II (e.g. Taylor series, big-O notation, asymptotic analysis)
13. Probability I (e.g. Gaussians, entropy and mutual information, KL divergence)
14. Probability II (e.g. Bayes' theorem, variational inference, Markov Chain Monte Carlo)
15. Linear programs I (e.g. primal and dual formulation, discrete geometry)
16. Linear programs II (e.g. simplex method, ellipsoid method)
17. Semidefinite and quadratic programs
18. Integer programs
19. Spanning trees, shortest paths
20. Maxflow

Outline

Outline

Outline

Outline

Outline

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)
26. Heaps

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)
26. Heaps
27. Hashing

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)
26. Heaps
27. Hashing
28. Huffman trees and suffix trees

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)
26. Heaps
27. Hashing
28. Huffman trees and suffix trees
29. Approximation algorithms I

Outline

21. Packing problems
22. Sorting
23. Simple data structures
24. Amortized cost
25. Advanced binary search trees (e.g. splay trees)
26. Heaps
27. Hashing
28. Huffman trees and suffix trees
29. Approximation algorithms I
30. Approximation algorithms II

Outline

Outline

Outline

Outline

31. Fast matrix multiplication and FFT
32. The probabilistic method
33. Compressed sensing

Outline

Outline

Outline

31. Fast matrix multiplication and FFT
32. The probabilistic method
33. Compressed sensing
34. Information theory
35. Graphical models
36. PAC Bayes

Outline

31. Fast matrix multiplication and FFT
32. The probabilistic method
33. Compressed sensing
34. Information theory
35. Graphical models
36. PAC Bayes
37. Basic neural networks

Outline

31. Fast matrix multiplication and FFT
32. The probabilistic method
33. Compressed sensing
34. Information theory
35. Graphical models
36. PAC Bayes
37. Basic neural networks
38. Deep learning theory I

Outline

31. Fast matrix multiplication and FFT
32. The probabilistic method
33. Compressed sensing
34. Information theory
35. Graphical models
36. PAC Bayes
37. Basic neural networks
38. Deep learning theory I
39. Deep learning theory II

Logistics

Logistics

- Classes MWF 4-5 pm (Montreal time)

Logistics

- Classes MWF 4-5 pm (Montreal time)
- Office hours tentatively Monday 5 pm and Friday 10 am

Logistics

- Classes MWF 4-5 pm (Montreal time)
- Office hours tentatively Monday 5 pm and Friday 10 am
- **All classes and office hours via Zoom**

Logistics

- Classes MWF 4-5 pm (Montreal time)
- Office hours tentatively Monday 5 pm and Friday 10 am
- **All classes and office hours via Zoom**
- All materials and Zoom links at MyCourses:
  `https://mycourses2.mcgill.ca/`

Grading

Student evaluations will be made as follows:

Grading

Student evaluations will be made as follows:

- 10%, attendance at class meetings

Grading

Student evaluations will be made as follows:

- 10%, attendance at class meetings
- 10%, making a diligent effort during participation in class (getting the “right answer” is not important, but thinking is)

Grading

Student evaluations will be made as follows:

- 10%, attendance at class meetings
- 10%, making a diligent effort during participation in class (getting the “right answer” is not important, but thinking is)
- 80%, performance on problem sets - the 5 or 6 problem sets will be weighted equally

Grading

Student evaluations will be made as follows:

- 10%, attendance at class meetings
- 10%, making a diligent effort during participation in class (getting the “right answer” is not important, but thinking is)
- 80%, performance on problem sets - the 5 or 6 problem sets will be weighted equally

Note: An exception may be made for the attendance/participation portions of the grade for those students who are attending the class remotely from a time zone that prohibits live participation. Such students must describe such circumstances to the instructor in advance.

Problem sets

Problem sets

- 5 or 6 problem sets, about every two weeks

Problem sets

- 5 or 6 problem sets, about every two weeks
- Important for learning the material

Problem sets

- 5 or 6 problem sets, about every two weeks
- Important for learning the material
- Proof-based - we will learn how to write those proofs

Problem sets

- 5 or 6 problem sets, about every two weeks
- Important for learning the material
- Proof-based - we will learn how to write those proofs
- Start them early! :)

Collaboration policy

Collaboration policy

- Collaboration is an important part of understanding

Collaboration policy

- Collaboration is an important part of understanding
- Talking to fellow students about the problem sets is encouraged

Collaboration policy

- Collaboration is an important part of understanding
- Talking to fellow students about the problem sets is encouraged
- Please list all fellow students you worked with

Collaboration policy

- Collaboration is an important part of understanding
- Talking to fellow students about the problem sets is encouraged
- Please list all fellow students you worked with
- However, written solutions must be completely your own

Collaboration policy

- Collaboration is an important part of understanding
- Talking to fellow students about the problem sets is encouraged
- Please list all fellow students you worked with
- However, written solutions must be completely your own
- **Do not share proofs** / **Overleaf** / **written notes etc.** with other students

Collaboration policy

- Collaboration is an important part of understanding
- Talking to fellow students about the problem sets is encouraged
- Please list all fellow students you worked with
- However, written solutions must be completely your own
- **Do not share proofs** / **Overleaf** / **written notes etc.** with other students
- If two collaborators submit material that looks suspiciously similar, both will get lower marks

LaTeX

LaTeX

- System for typesetting math

LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class

LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class
- **All problem set solutions except for the first must be written in LaTeX and submitted as PDFs**

LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class
- **All problem set solutions except for the first must be written in LaTeX and submitted as PDFs**
- A template doc for solutions will be provided (also the source LaTeX for problem sets / slides)

LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class
- **All problem set solutions except for the first must be written in LaTeX and submitted as PDFs**
- A template doc for solutions will be provided (also the source LaTeX for problem sets / slides)
- Figures need not be created in LaTeX, though they should be included in the PDF

LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class
- **All problem set solutions except for the first must be written in LaTeX and submitted as PDFs**
- A template doc for solutions will be provided (also the source LaTeX for problem sets / slides)
- Figures need not be created in LaTeX, though they should be included in the PDF
- LaTeX mistakes (if still readable) will not result in a lower grade

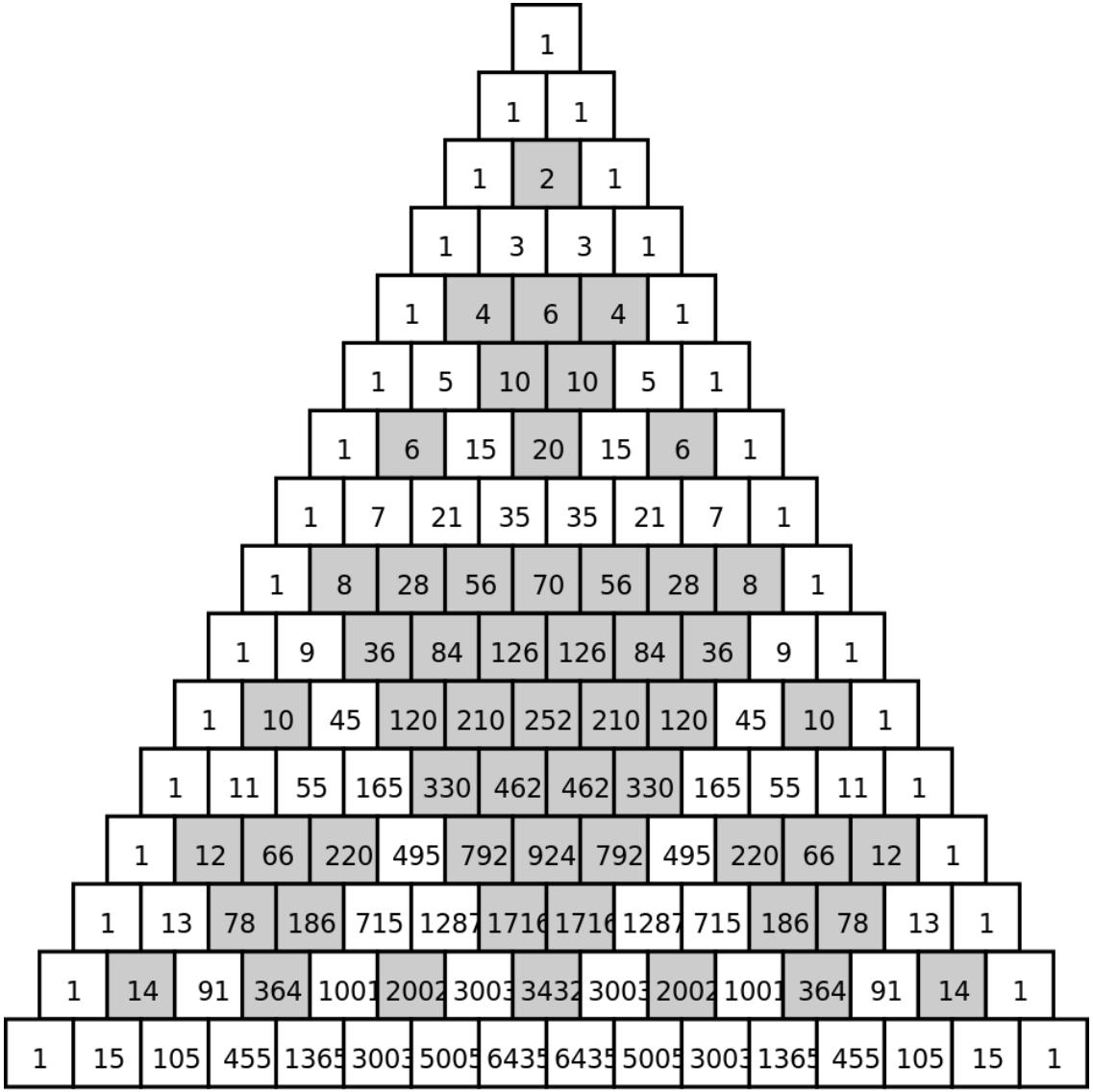
LaTeX

- System for typesetting math
- Good skill for paper-writing etc, independent of this class
- **All problem set solutions except for the first must be written in LaTeX and submitted as PDFs**
- A template doc for solutions will be provided (also the source LaTeX for problem sets / slides)
- Figures need not be created in LaTeX, though they should be included in the PDF
- LaTeX mistakes (if still readable) will not result in a lower grade

Example resources for learning LaTeX:

- `https://www.overleaf.com/learn/latex/Free_online_introduction_to_LaTeX_(part_1)`
- `https://artofproblemsolving.com/wiki/index.php/LaTeX`

Questions?


1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 186 715 1287 1716 1716 1287 715 186 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1