Lecture 1

Thinking like a Computer Scientist

Chinmay Nirkhe | CSE 421 Spring 2026

Thinking like a Computer Scientist

The computational lens

Nobel Physics Prize Awarded for Pioneering A.I. Research by 2 Scientists

With work on machine learning that uses artificial neural networks, John J. Hopfield and Geoffrey E. Hinton "showed a completely new way for us to use computers," the committee said.

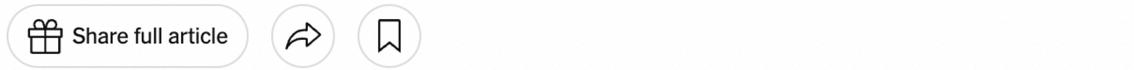
The New York Times, October 2024

Nobel Prize in Chemistry Goes to 3 Scientists for Predicting and Creating Proteins

The Nobel, awarded to David Baker of the University of Washington and Demis Hassabis and John M. Jumper of Google DeepMind, is the second this week to involve artificial intelligence.

The New York Times, October 2024

Ideas & Trends; The Nobels: Dazzled By the Digital Light



By George Johnson

Oct. 15, 2000

The New York Times, October 2000

There is no Nobel Prize for computer science. But obliquely and perhaps unconsciously, the judges were using the tools at their disposal to recognize how formidable the notion of information has become, pervading not just the technologies we devise but the way we think about ourselves.
— George Johnson

Thinking like a Computer Scientist

The computational lens

Nobel Physics Prize Awarded for Pioneering A.I. Research by 2 Scientists

With work on machine learning that uses artificial neural networks, John J. Hopfield and Geoffrey E. Hinton "showed a completely new way for us to use computers," the committee said.

The New York Times, October 2024

Nobel Prize in Chemistry Goes to 3 Scientists for Predicting and Creating Proteins

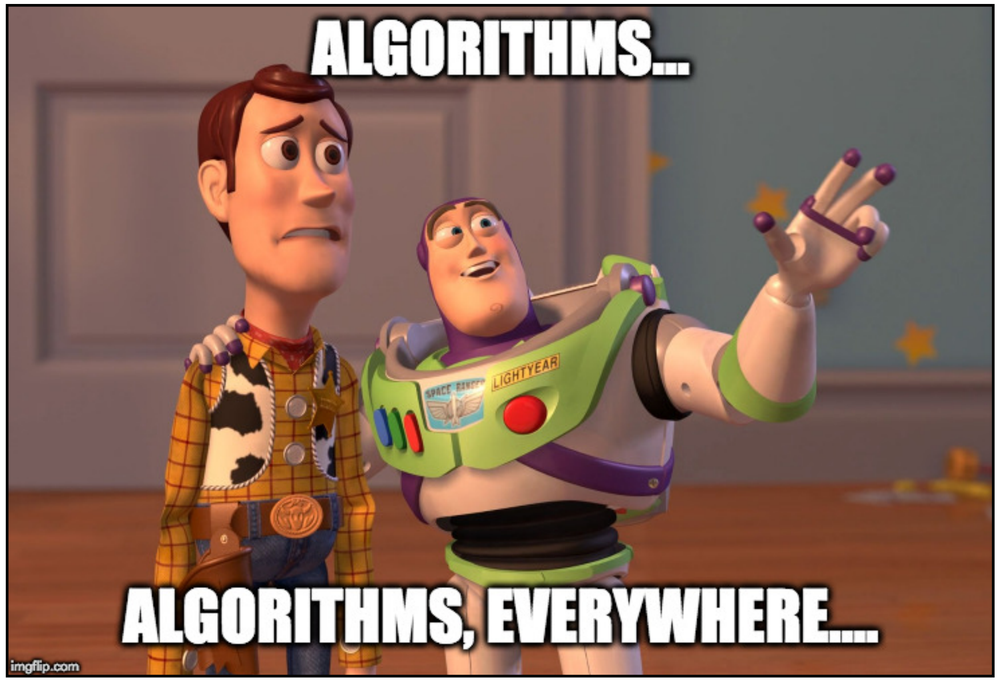
The Nobel, awarded to David Baker of the University of Washington and Demis Hassabis and John M. Jumper of Google DeepMind, is the second this week to involve artificial intelligence.

The New York Times, October 2024

- 2024 seemed like it was the year of "thinking like a computer scientist". But really its the 2000s that is the century of thinking like a computer scientist.
- As the problems we wish to solve get bigger and bigger, understanding the *computational cost* associated with solving problems will gain an outsized importance.
- My goal is to train you how to think about the world through a *computational lens*.

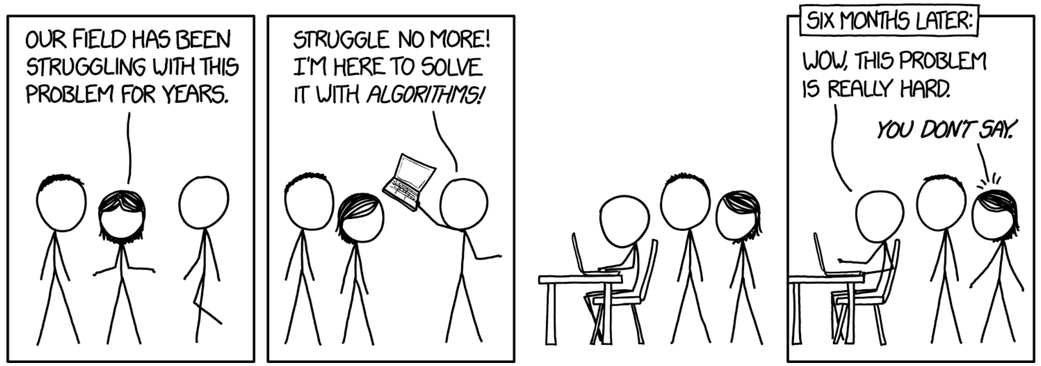
My goals for you in this course

- Help you learn to identify algorithmic problems
- Develop a toolkit for finding efficient algorithms
- Develop techniques for proving *correctness* and *analyzing* their properties
- Communicate your algorithms and their properties to others
- **New!** Learn to use AI as a tool for developing algorithms

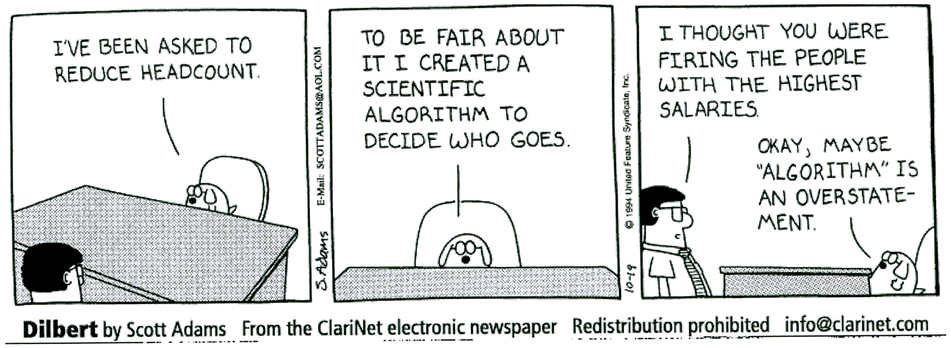


Properties of an algorithm

What should we be optimizing for?



- Computational efficiency
  - Speed, time, communication, etc.
  - Historically, important parameters because computers were slow and weak
  - Today, important parameters because computations are large
- Correctness
  - How to prove that an algorithm does what it intends on *all inputs*
- Fairness
  - Does my algorithm have unintended consequences in its optimization?
  - Incredibly important as we apply algorithms in the real world
- In my research world of quantum computing
  - How much entanglement does my algorithm generate?
  - Is my algorithm error-robust? What kind of errors can it tolerate?



Course Logistics

Instructor

Chinmay Nirkhe [he/him]
nirkhe@cs.washington.edu

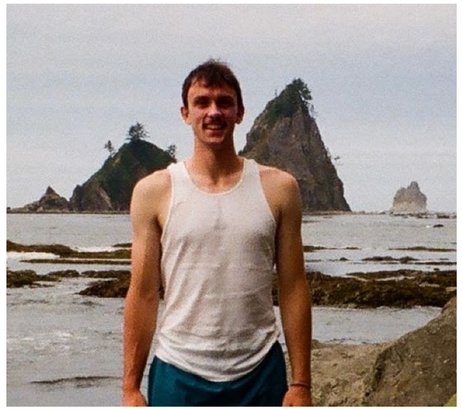
Speciality: Complexity, Quantum
Office: CSE2 Room 217
Office Hours: Wednesdays 2:30 - 4:00pm

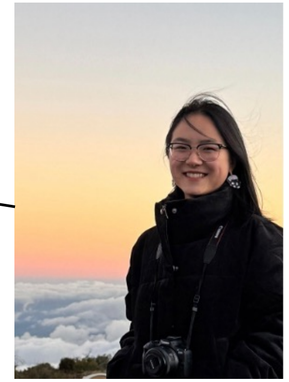
The course staff

- **Head TA:** **Toby**

  Contact Head TA with any questions, logistics, illnesses, etc.

- **TAs:** Shreya, Nicos, Toby, Jay, Shayla





421, one iteration to the next

- This is my second time lecturing this course.
- Significant recalibration and reorganization was made since the first iteration.
- This second iteration is redesigned to:
  - Make more clear what is expected to achieve a 3.0/3.5/4.0 etc.
  - Provide opportunity for talented students to go beyond the minimum course material and really understand the theory of algorithms
  - Integrate generative AI as a tool for *enhancing* learning

How to succeed in this class

- Attend lecture, sections, and ask questions/interact. Attendance is correlated strongly with final grade.
- I remember who asks questions, attends my office hours, and participates.
  - Can make the difference when I'm assigning grades if you are on the cusp.
  - It helps if I know who you are if you want me to write you a recommendation letter for a job or graduate school.
- Take lots of thinking walks and mull on these problems. *Being a computer scientist is different than being a programmer*. The same techniques may not apply.

CSE 421 Section

- Section is a great way to get practice on the week's topics
- Section problems are designed to
  - help you learn the material
  - help you learn how to write algorithms/solutions effectively
- While problem sets are designed to help you evaluate how well you have learned the material (a regularizer, so to speak)
- Section problems are often correlated with problem set problems as well :)

Coursework

- Problems are due on Fridays at 6:00 PM
  - Each week, 3 general questions and 1 challenge question.
  - Week 4 has 1 general question and 1 challenge question.
- See the homework guide on the website.
- It is acceptable to talk to other students but write-ups must be done individually and *cannot* be shared/distributed.
- AI use is permitted (in specific moderation)! **Read the syllabus, AI guide and problem set guide**. Ask TAs if you have *any* questions.

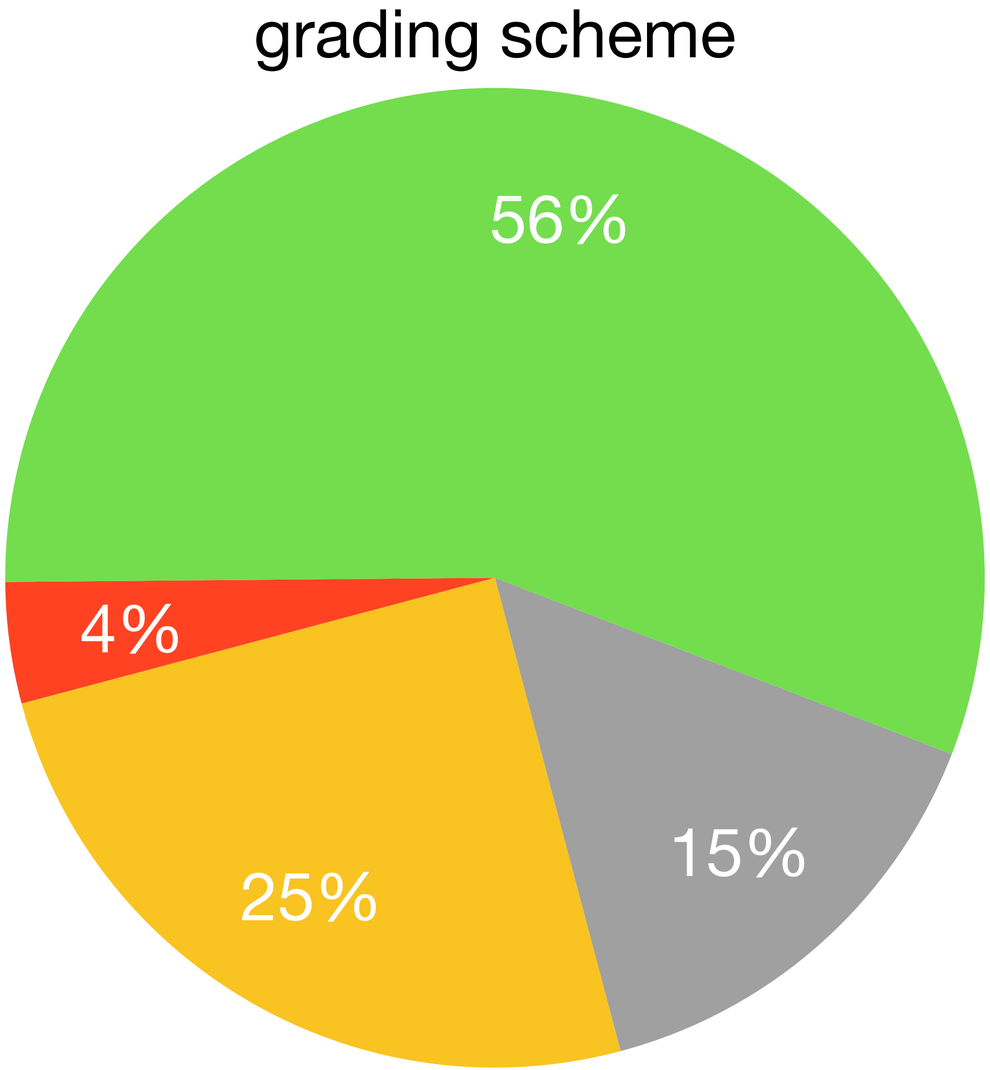
Late Problem Sets

15 late problems

- No reason required!
- Extension is for 24 hours. Only 1 extension per problem.
- Extensions are **per problem**. You can use extensions on any number of problems per set.
- Problem is given a 0 after 24 hours.
- Extenuating circumstances can be discussed with Chinmay; no TA can approve additional extensions.
- **However, TAs will not be available for help after the original deadline.**

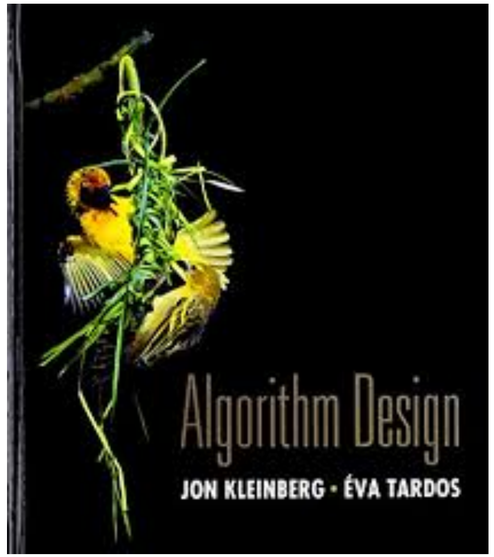
Exams and grading scheme

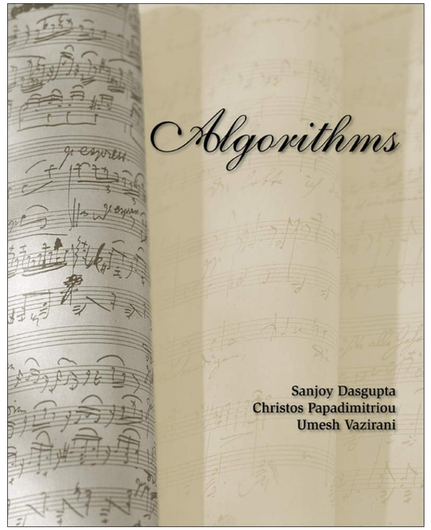
- The dates are set. Due to the size of the class, no late exam exceptions will be made. Extenuating circumstances should be discussed with Chinmay only.
- Midterm: February 2nd 5:30-7:20pm
- Final: March 16th 2:30-4:20pm
- Grading scheme:
  - Top 22 general problems (25 total): 56%
  - Midterm*: 15%
  - Final*: 25%
  - Top 7 challenge questions (9 total): 4%
- .92 guarantees a 4.0, .75 guarantees a 3.0



Textbook

- There are two suggested textbooks for the course.
- Both are page-turners and are great for learning how to think like an algorithm designer.
- They are also not required — all required content can be extracted from the lecture notes and quiz section materials.





421 coursework

General problem set questions

- Problem sets will this iteration have 3 questions and a challenge question.
- General problem set questions account for 56% of the grade.
  - Designed to test understanding of lecture material.
  - Each question *will* require using a concept from lecture in a new manner.
  - Some amount of struggle will be necessary to solve them, but the answers will never be unduly long.
  - You can get a 4.0 in the course with strong exam scores and answering only the general problem set questions.
  - You are graded on both the correctness of the solution and writing quality.

421 problem sets

Challenge problem set questions

- Problem sets will this iteration have 3 questions and a challenge question.
- Challenge problem set questions account for 4% of the grade.
  - Challenge questions are designed to be solvable by all students in this course.
  - However, they will require you to think deeply about the concepts and apply them in a new way.
  - Challenge questions are the questions which generative AI struggles on!
  - Challenge questions are more difficult than exam questions.
  - If you want a recommendation letter from me, solve the majority of the challenge questions!

421 exam questions

- Exam questions are not designed to trick you.
- They are direct variations of questions you have seen in problem sets, section, or lecture.
- Additionally, they have been modified to minimize writing and test you on concepts directly.
- Practice exams will be provided with solutions.
- Average grades around 75-85% are expected on exams.
- A curve function of $x \mapsto \frac{\log(1+\kappa x)}{\log(1+\kappa)}$ may be applied with $\kappa = 2$.

AI use policy

- Over the past year, I have come to use generative AI more and more in my research.
- It is an invaluable tool that can assist with learning and organizing thoughts, researching a multitude of techniques, etc.
- I **want** you to have the autonomy to learn in whatever manner best fits you!
- AI will be a great tool for many of you and this class will help you both learn the theory of algorithms and
- It also has the potential to detract from learning.

AI use policy

- I’ve developed a specific GPT for this course.
  - It has been designed to provide small hints and has been specifically designed for the problem sets and exams of this course.
  - However, it is no substitute for our TAs because it *lacks* contextually. I highly recommend you only the GPT when truly stuck.
  - It is designed to help with small writing questions, but will not rewrite your whole solution for you.
- You are allowed to use *only* this GPT (and no others) to assist you.
- The prerequisites for use are that
  - You have tried the problem for some time (~30 min) by yourself or with collaborators
  - Or you have visited a TA or me for guidance before you use the GPT.

421 problem set and exam correlation

- In past years, problem set scores correlated well with exam success and also with a comprehensive understanding of the *theory of algorithms*.
- This term, I suspect, with the AI adjustment and the challenge problems, the correlation between problem set scores and exams will be *less*.
- Instead, there will be a stronger correlation between *how you solve* problem sets and exam scores.
- Take the time to solve problem sets in a way that maximizes your learning! I'm entrusting you to figure out what works best for you.

The theory of algorithms

- In this course you will learn many algorithmic techniques as well as how to prove that the algorithms are *correct*.
- Much of the emphasis of this course will be on proving that your algorithm is correct! Why emphasize this?
  - Just checking that your algorithms works on a few examples is not enough.
  - Real-world scenarios are hard to anticipate. Instead, a proof of correctness encompassing *all* possible instantiations is best.
  - In your future jobs, you might be asked to prove that an algorithmic implementation will do its job. It's your responsibility to be able to do that well.
  - This is where gen-AI struggles (in my experience)! It's great at conjecturing algorithms but struggles to prove that they work correctly. Human + gen-AI is awesome here because you have the contextually to understand the problem and ensure that your solution is correct.

Day 1 checklist

- Find the course website. https://courses.cs.washington.edu/courses/cse421/
- **Read the syllabus and read the AI tools policy.**
- Make sure you are on the following course resources: Gradescope, EdStem. Ask a TA for help, if you are not.
- **Attend your first section this week.**
- Problem set 1 is posted soon. You have the knowledge to start after section.
- Get all your credit: 4th credit available by signing up for 490D.