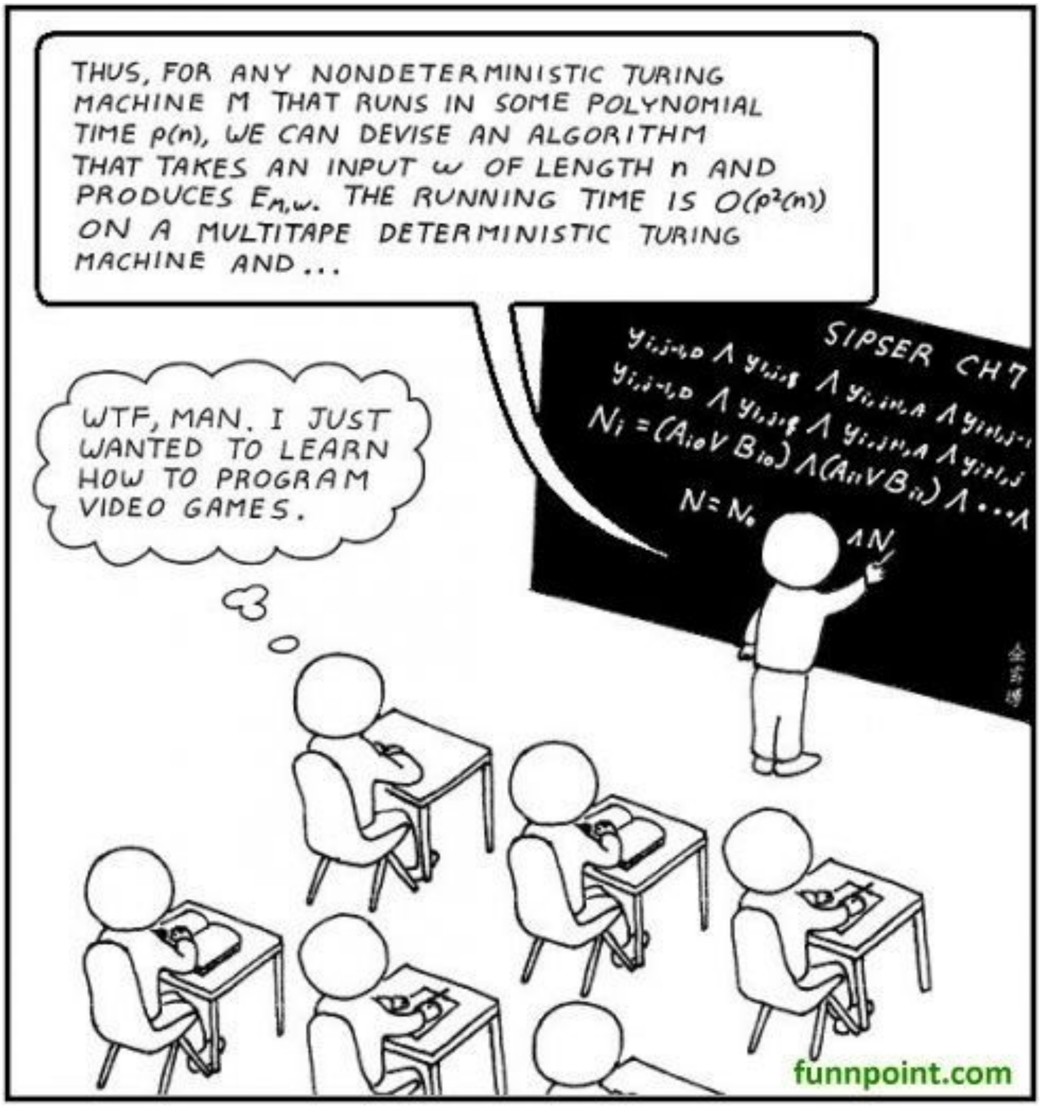
CSCI 3202

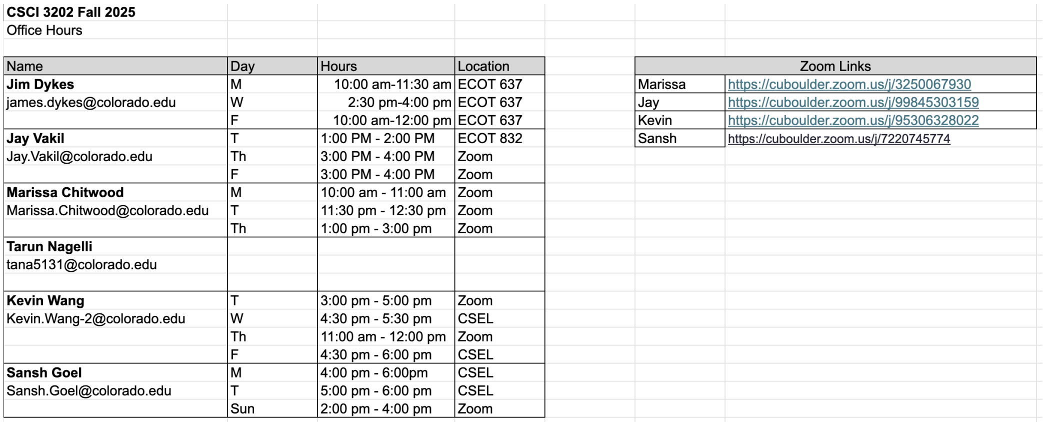
Lecture 7

September 8, 2025


THUS, FOR ANY NONDETERMINISTIC TURING MACHINE M THAT RUNS IN SOME POLYNOMIAL TIME $p(n)$, WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT w OF LENGTH n AND PRODUCES $E_{M,w}$. THE RUNNING TIME IS $O(p^2(n))$ ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...
WTF, MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.
SIPSER CH7
$N_i = (A_{i0} \vee B_{i0}) \wedge (A_{i1} \vee B_{i1}) \wedge \cdots \wedge$
$N = N_0$
$\wedge N$
funnpoint.com

Announcements

- Office Hours are posted on Canvas. Please ask for help if you need it



CSCI 3202 Fall 2025
Office Hours

Name	Day	Hours	Location
Jim Dykes	M	10:00 am-11:30 am	ECOT 637
james.dykes@colorado.edu	W	2:30 pm-4:00 pm	ECOT 637
	F	10:00 am-12:00 pm	ECOT 637
Jay Vakil	T	1:00 PM - 2:00 PM	ECOT 832
Jay.Vakil@colorado.edu	Th	3:00 PM - 4:00 PM	Zoom
	F	3:00 PM - 4:00 PM	Zoom
Marissa Chitwood	M	10:00 am - 11:00 am	Zoom
Marissa.Chitwood@colorado.edu	T	11:30 pm - 12:30 pm	Zoom
	Th	1:00 pm - 3:00 pm	Zoom
Tarun Nagelli			
tana5131@colorado.edu			
Kevin Wang	T	3:00 pm - 5:00 pm	Zoom
Kevin.Wang-2@colorado.edu	W	4:30 pm - 5:30 pm	CSEL
	Th	11:00 am - 12:00 pm	Zoom
	F	4:30 pm - 6:00 pm	CSEL
Sansh Goel	M	4:00 pm - 6:00pm	CSEL
Sansh.Goel@colorado.edu	T	5:00 pm - 6:00 pm	CSEL
	Sun	2:00 pm - 4:00 pm	Zoom

Zoom Links	
Marissa	https://cuboulder.zoom.us/j/3250067930
Jay	https://cuboulder.zoom.us/j/99845303159
Kevin	https://cuboulder.zoom.us/j/95306328022
Sansh	https://cuboulder.zoom.us/j/7220745774

- Homework 2 due on Wednesday, Sept 10 by 11:59 pm
    - In addition to util4e.py, you will need to copy util.py to make your code work

Homework 2

- Create a data structure to hold the graph. I would use a dictionary
- Define a graph_problem subclass of class Problem and fill in the needed functions.
- graph_problem should use a data structure that holds your graph
- In addition to `utils4e.py`, you will need to copy `utils.py` into your local directory
- Visit the nodes in ascending order only when you have to choose between several nodes, such as when visiting the children of a node
    - You may need to sort parts of your data structure to make this work with the AIMA code
- Return the optimal path for BFS. The path found by DFS may or may not be optimal. Return the path found by DFS

Quiz #2

- Average was 73%
- Missed questions about BFS and DFS characteristics
- Review

Readings

- AIMA sections 3.4-3.6 for today
- Subjects: UCS, Greedy, A*

Lecture

- UCS Search
  - Uninformed
  - Weighted BFS Search
  - Uses Priority Queue with path cost so far as the metric
  - Path Cost is the sum of the individual step costs for a path
- A* Search
  - Informed search
  - A* search adds a heuristic that provides an estimated distance to the goal
  - We add the path cost to node n to the heuristic for node n to the goal to get the estimated distance
  - Use this estimated distance in UCS
  - This can greatly reduce the number of nodes visited to construct an optimal path
- Greedy Search
  - Informed search
  - Defines a heuristic function
  - Heuristic is distance from a node to the goal (straight line)
- Ucs greedy astar compare.pdf
- Properties of AStar
  UCS Greedy and AStar Slides.pdf

Path

- For a GPS system, the path is the important item most people want. How do you construct the path?
- Start at the end and work backwards
  - Start at destination
  - Find the most recent node that explored the destination
  - Mark this node as being on path
  - Repeat with this node as destination
  - Done when you reach the starting node
- Often implemented as a Python dictionary
  - Each node is a key
  - Value is node that visited or explored it
  - If there are multiple explorations, dictionary only keeps most recent one

Next Class

- Heuristics
- HW 3 Released on Wednesday