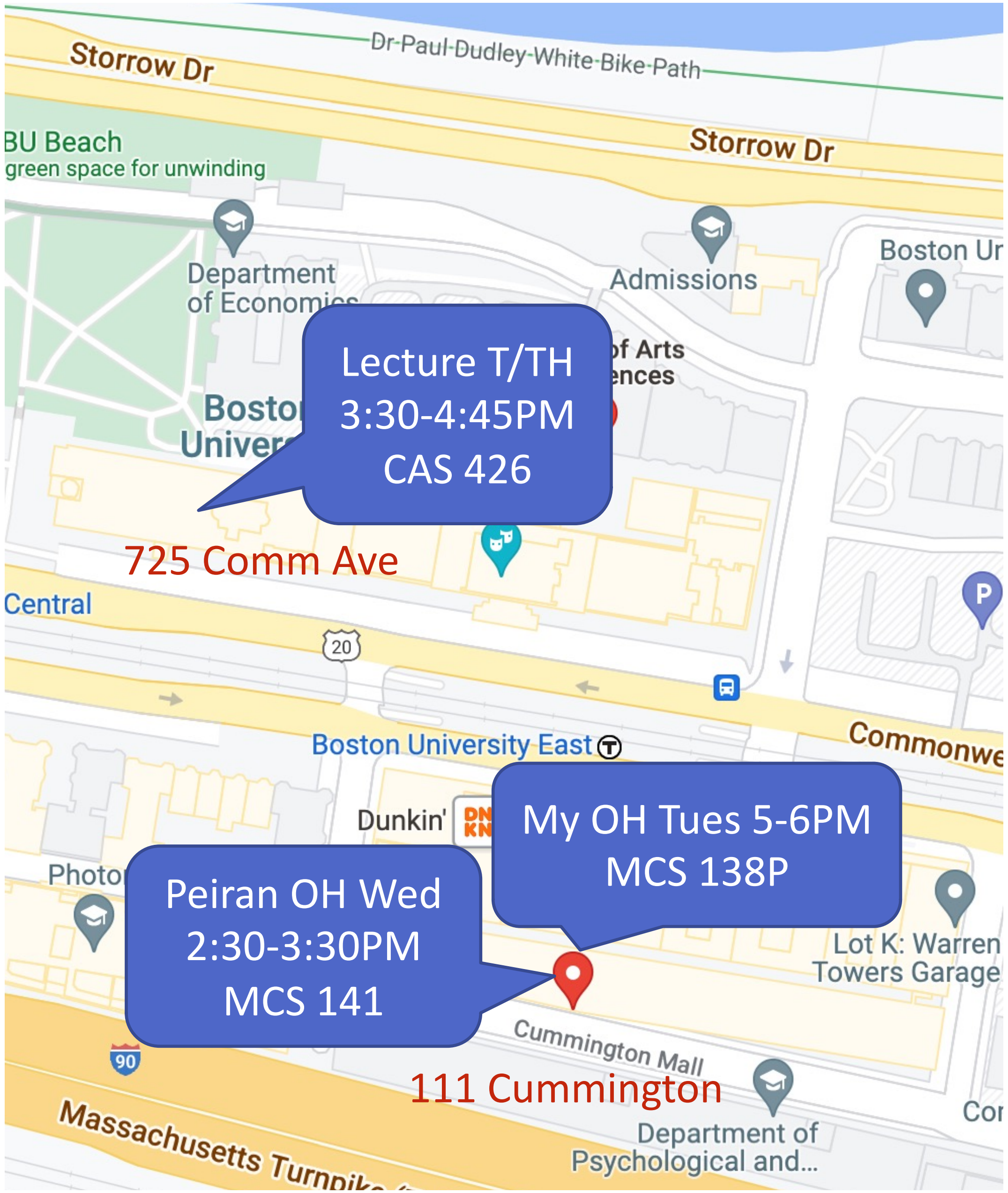
DS 574: Algorithmic Mechanism Design

PROFESSOR KIRA GOLDNER

Teaching Staff

Instructor: Prof. Kira Goldner
Email: goldner@bu.edu
OH: Tuesday 5-6PM and by appointment
Office Location: 111 Cummington Mall, 138P

TF: Peiran Xiao
Email: pxiao@bu.edu
OH: Wednesday 2:30-3:30PM and by appt
Location: 111 Cummington Mall, 141



Introductions!

- Name
- Department + Year
- Why are you taking this class?
- Favorite pasta shape?

Class Resources

Course website: https://www.kiragoldner.com/teaching/DS574/

- Lecture notes, links to everything

Piazza (access code AMD):

- Questions and answers; alternative for email
- I am a human who does not live inside the computer!

Gradescope (entry code DJP34R):

- Turn in assignments and view grades

Sign up for these if you have not already! (Links on… the course website!)

Also! I am open to suggestions on how to best utilize things like Piazza!

This is a theoretical problem-solving class

No programming assignments! Evaluation based on problem sets and project.

Prerequisites:

- A first proofs class that's Discrete-Math-esque (DS 122, CS 131, MA 293, …)
- Undergrad algorithms (DS 320, CS 330, …)—algorithmic reasoning, runtime and complexity notions
- Intro probability (MA 581)—know r.v.s and compute their moments
- Mathematical maturity

Not expected:

- Any background in game theory/incentives/economics.

Evaluation

Homework (45%)

- Collaborative problem sets ~every other week.

Mechanism Design for Social Good problem formulation (15%)

- Formulate a problem and defend why the question is important both for the domain and within mechanism design. Identify a domain expert for potential collaboration.

Class participation (5%)

- In class and via Piazza (asking and answering questions) gets 100% here.

Final Project (35%)

- Investigate a research question not covered in class—read papers and write a survey OR do original research. Write up and presentation.

Homework Policies

- Expect to spend at least 10 hours per assignment.
- **Late policy:** You have 4 late days, max 2 per assignment (integer numbers used only). No exceptions.
- Type up homework with **LaTeX**.
- Turn in via **gradescope**. Due at 11:59pm on the date assigned.

- **Regrades:** Requests within 7 days, only via gradescope, with explanation/argument. Only for **incorrect** grading (not insufficient credit). If you request a regrade, the whole assignment/exam may be regraded, and your score may go up or down.

Collaboration Policy

Collaboration is encouraged!!!

- You may work with up to two classmates on an assignment. **List your collaborators' names on your assignment. (E.g., Collaborators: None.)**
- Good rough rule: Nobody should leave the room with anything written down. If you really understand, you should be able to reconstruct it on your own.
- You may **not** use the internet on homework problems. You may use course materials and the recommended readings from textbooks.

I believe **strongly** in learning over evaluation, learning via collaboration, and academic integrity. Please adhere to BU's academic conduct policy.

Class Etiquette

I strive toward an accessible and equitable classroom for all students.

- Raise your hand.
- Be conscious of how often you participate (in class and in collaboration).
  - Don't talk over others, leave room for other voices if you speak up a lot, and speak up more if you do not.
- I'm always open to new strategies here.

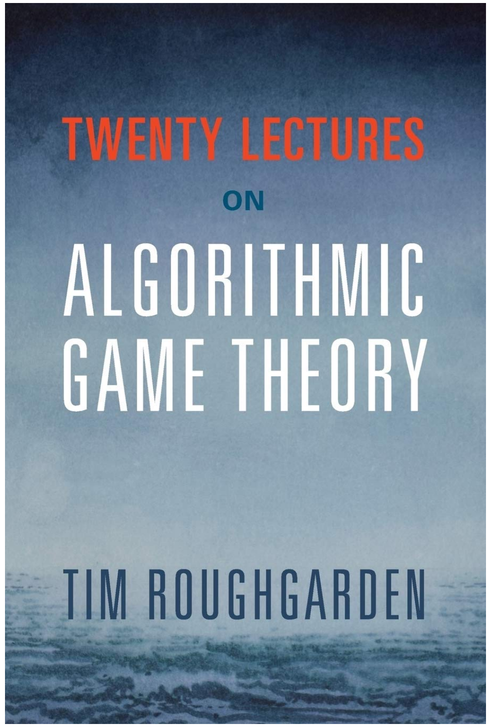
But also

- Ask questions!!!!!!

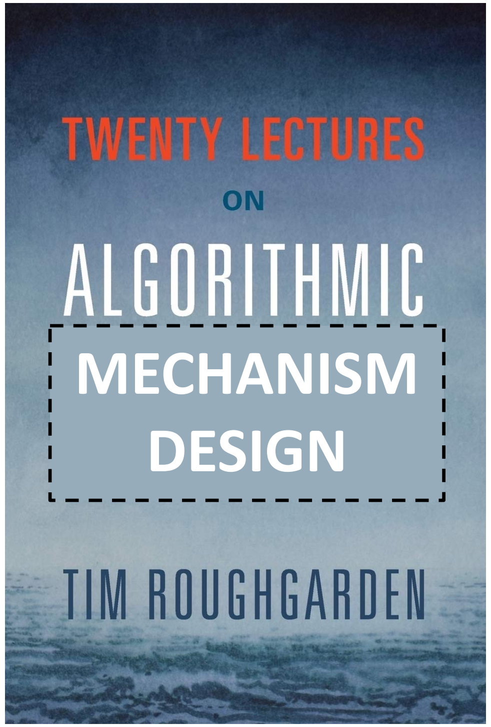
Best advice I ever got was to just ask and not wait to fill in gaps myself later.

Book

There is no required textbook, and the lecture notes will be self contained.
But many of the topics we are covering are well covered in standard algorithms textbooks; some lectures are adapted from Tim Roughgarden's lecture notes.



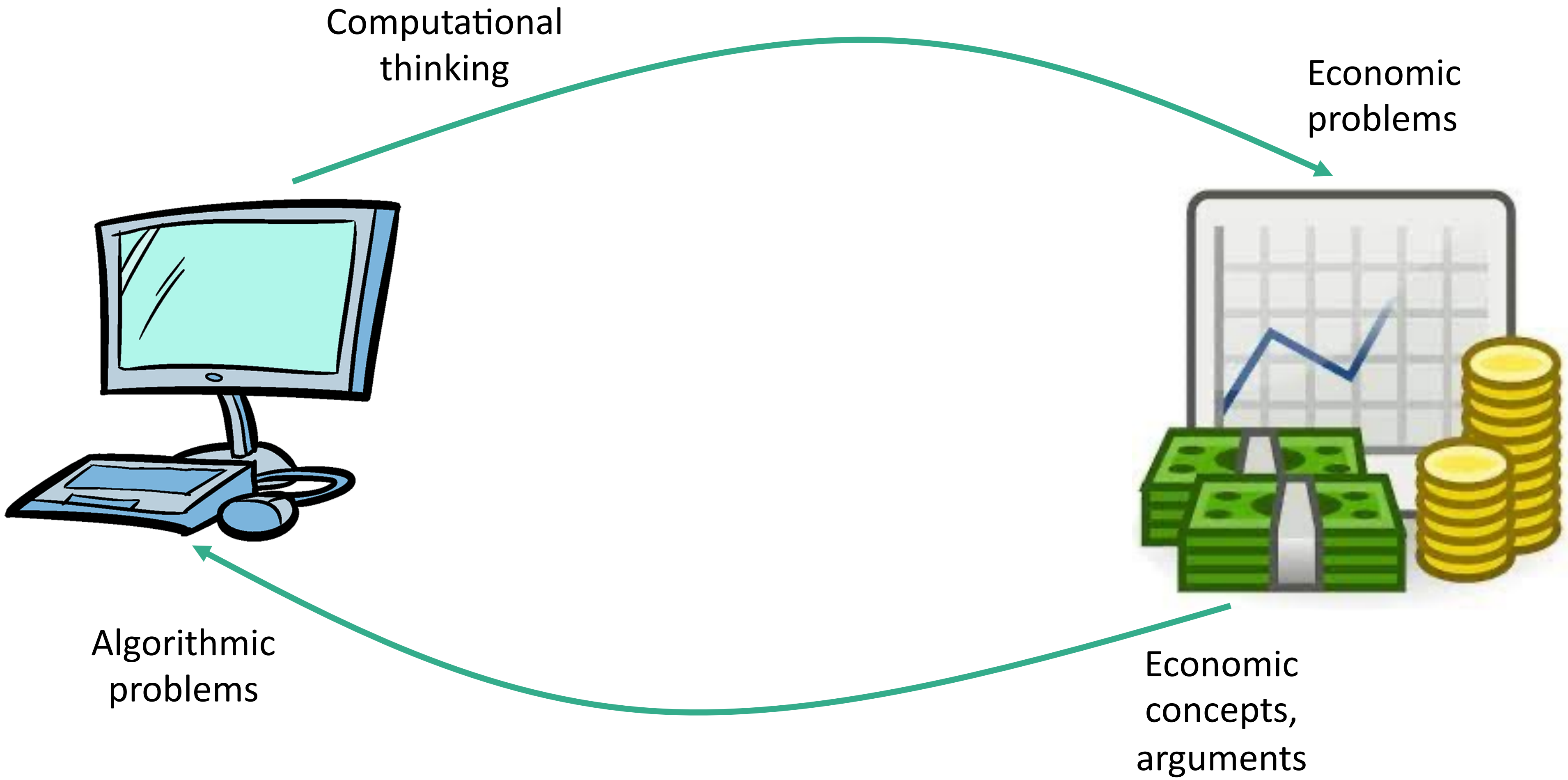
TWENTY LECTURES
ON
ALGORITHMIC
GAME THEORY
TIM ROUGHGARDEN



TWENTY LECTURES
ON
ALGORITHMIC
MECHANISM
DESIGN
TIM ROUGHGARDEN

What is “EconCS”?

Also referred to as:
Algorithmic Game Theory (AGT)



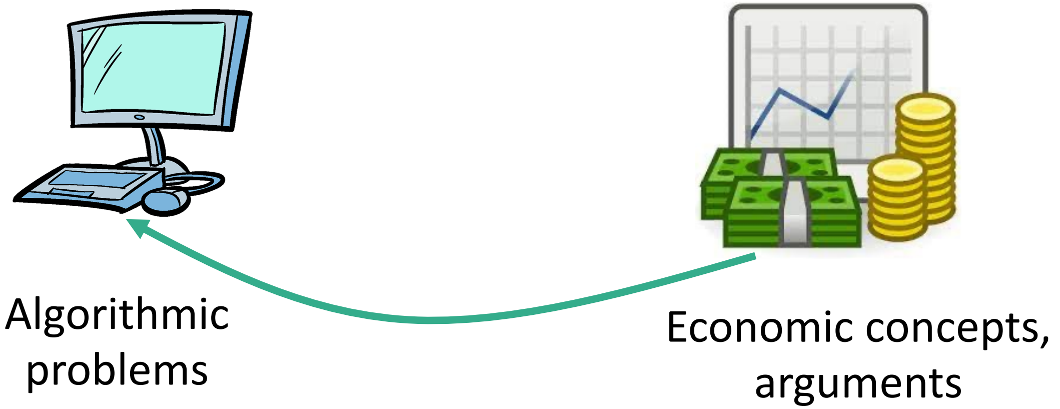
Computational thinking

Economic problems

Algorithmic problems

Economic concepts, arguments

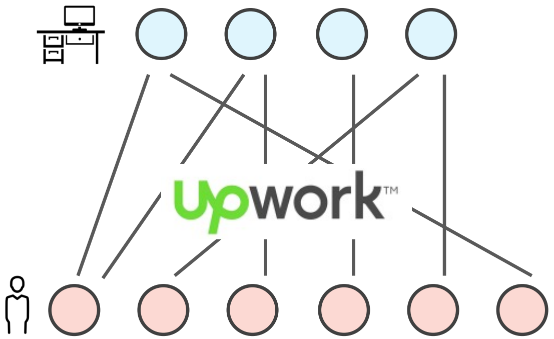
Econ→CS



Algorithmic problems

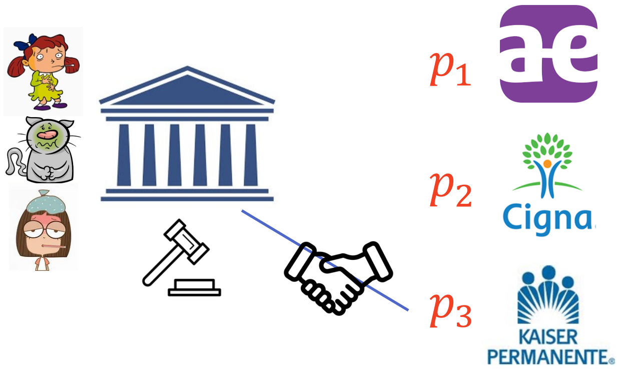
Economic concepts, arguments

Online Labor Markets

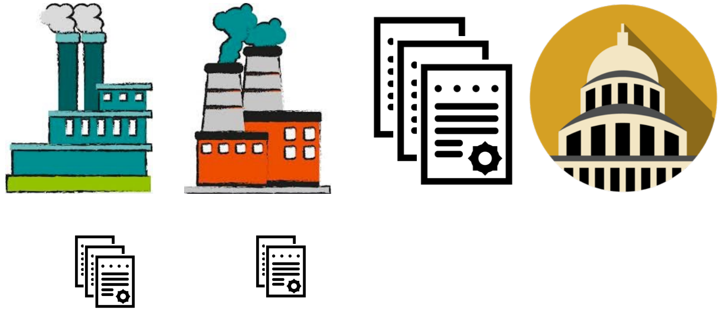


- The systems interact with **strategic individuals**.
- We must **design** them to be **robust** to **strategic behavior**.

Health Insurance



Carbon Emissions



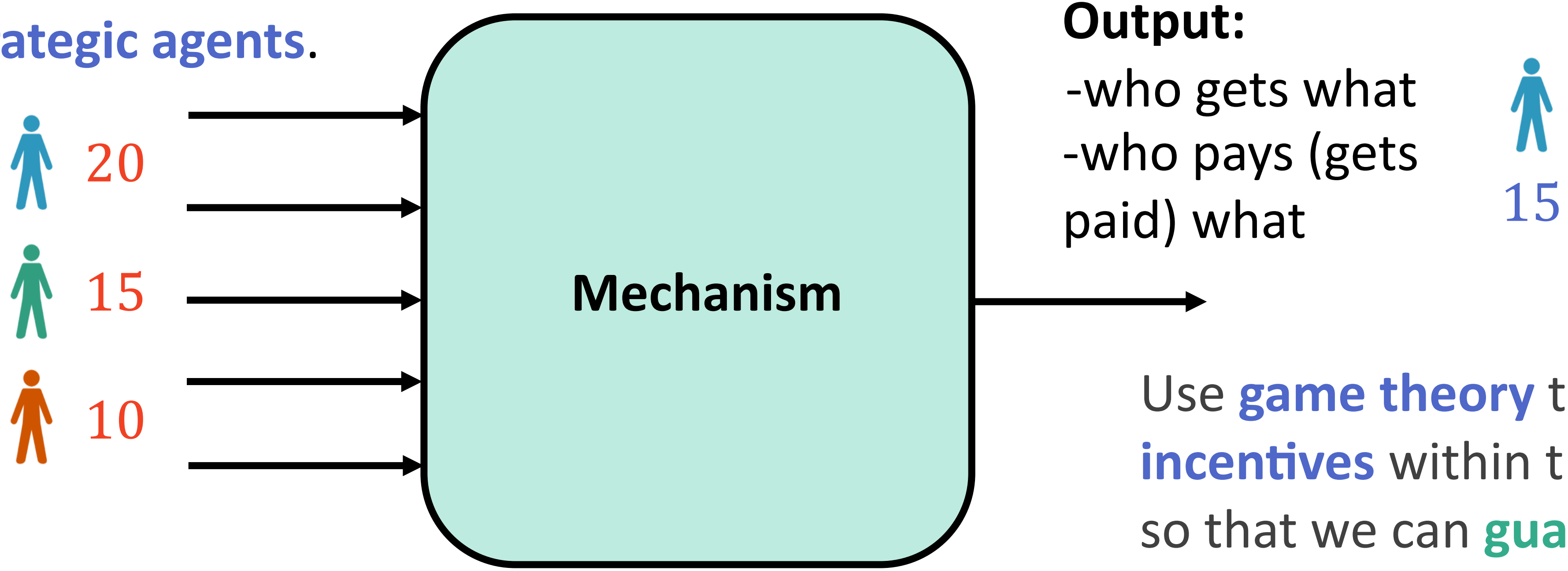
Econ→CS

Algorithmic problems

Economic concepts, arguments

Input:
Data reported by **strategic agents**.

Objective: Maximize buyer’s value

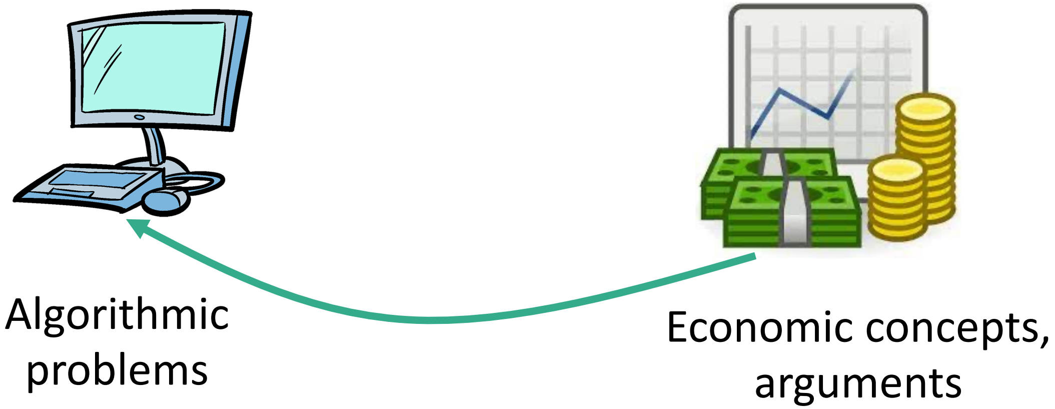


Output:
-who gets what
-who pays (gets paid) what

Use **game theory** to reason about **incentives** within the **algorithm** so that we can **guarantee** (approximate) optimality.

Econ→CS

Elegant **proofs** using an economic lens:



Algorithmic problems

Economic concepts, arguments

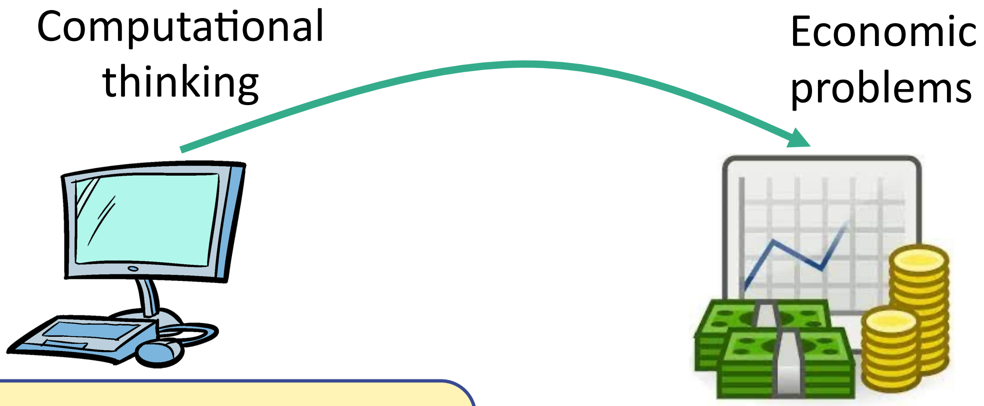
Maximum weight matching [Demange Gale Sotomayor ‘86]

- LHS runs ascending auction “bidding” on RHS until perfecting matching achieved.

Online bipartite matching [Karp Vazirani Vazirani ‘90]

- Algorithm: Randomly permute RHS. LHS arrives and takes first available item in LHS according to permutation.
- Prove this using elegant random price argument. [Eden Feldman Fiat Segal ‘21]

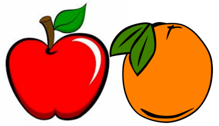
CS→Econ



1 item

Open Problem: What optimal mechanisms **can** we characterize beyond 1 item?

2 items



- Simple.
- Easy to compute.
- Only one real option.

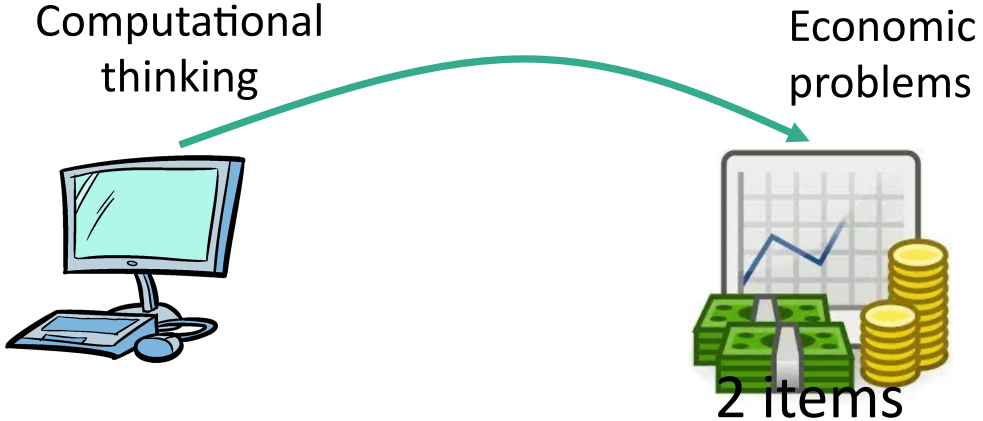
[Myerson '81]

$5: Pr[🍎]=1

- Uncountably infinite options. [Manelli Vincent '07, Daskalakis Deckelbaum Tzamos '15]
- Intractable to compute. [Daskalakis Deckelbaum Tzamos '13]
- We still know very little about how to do this.

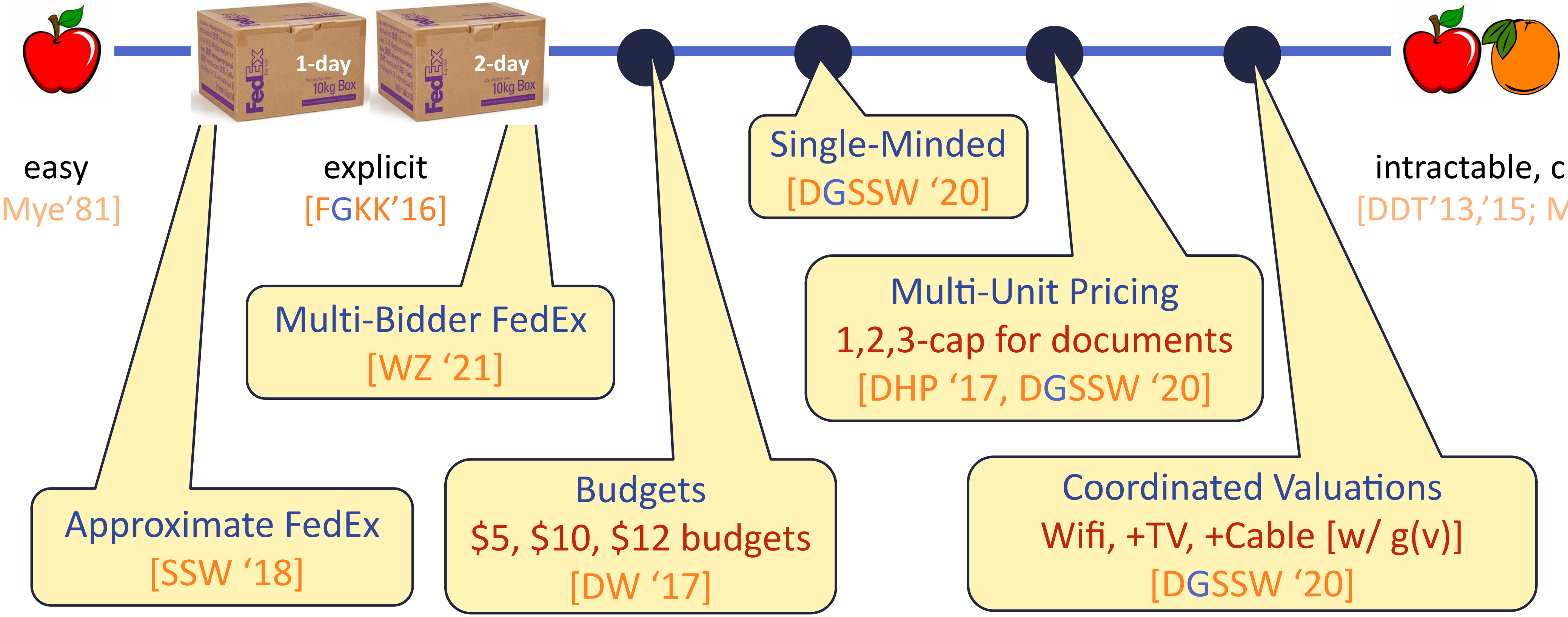
$5.89: (Pr[🍎]=.60, Pr[🍊]=.29)

CS→Econ



Computational thinking

Economic problems



1 item

easy
[Mye'81]

FedEx

explicit
[FGKK'16]

2 items

intractable, chaos
[DDT'13,'15; MV'07]

Approximate FedEx
[SSW '18]

Multi-Bidder FedEx
[WZ '21]

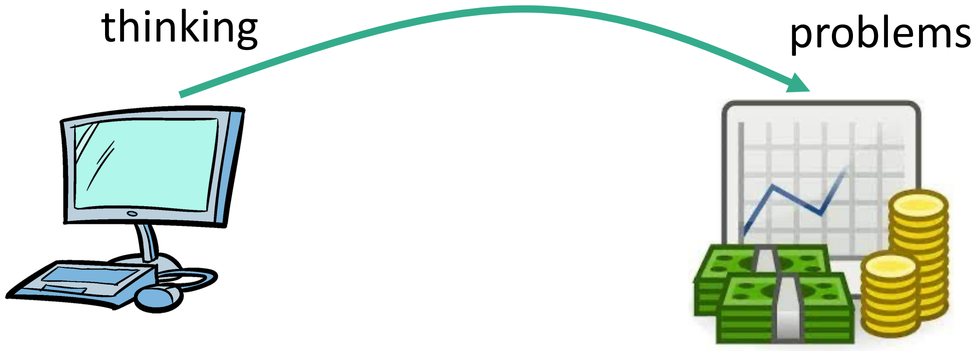
Budgets
$5, $10, $12 budgets
[DW '17]

Single-Minded
[DGSSW '20]

Multi-Unit Pricing
1,2,3-cap for documents
[DHP '17, DGSSW '20]

Coordinated Valuations
Wifi, +TV, +Cable [w/ g(v)]
[DGSSW '20]

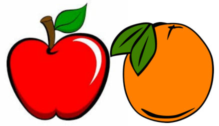
CS→Econ



Computational thinking

Economic problems

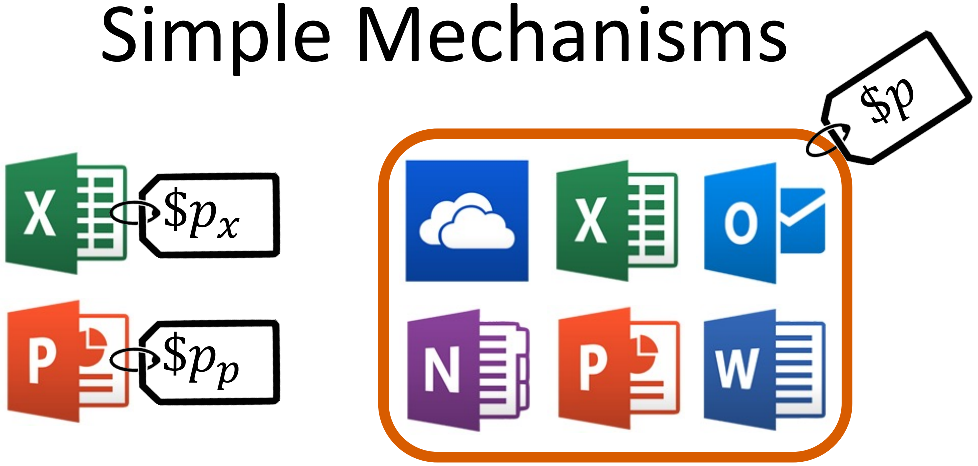
2 items



- Uncountably infinite options. [Manelli Vincent ‘07, Daskalakis Deckelbaum Tzamos ‘15]
- Intractable to compute. [Daskalakis Deckelbaum Tzamos ‘13]
- We still know very little about how to do this.

$5.89: (Pr[🍎]=.60, Pr[🍊]=.29)

Simple Mechanisms



(Lack of) information

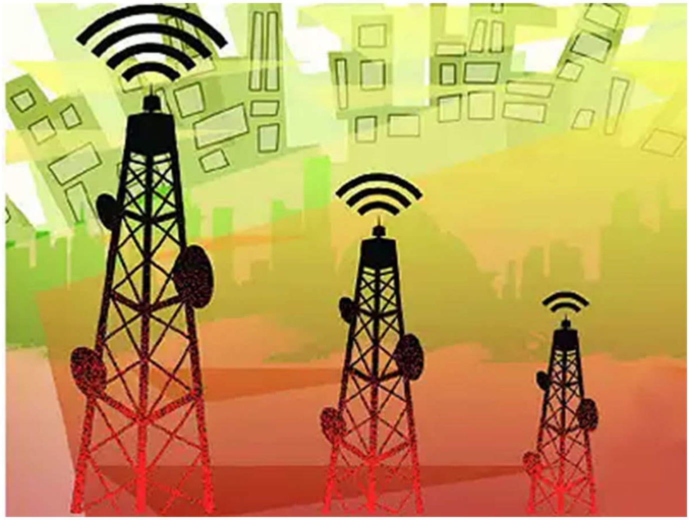
Robustness

Why is this important to learn about?

Mechanism Design and Society

Computationally Efficient:

- To design.
- To run.
- To strategize within.



Mechanism Design and Society

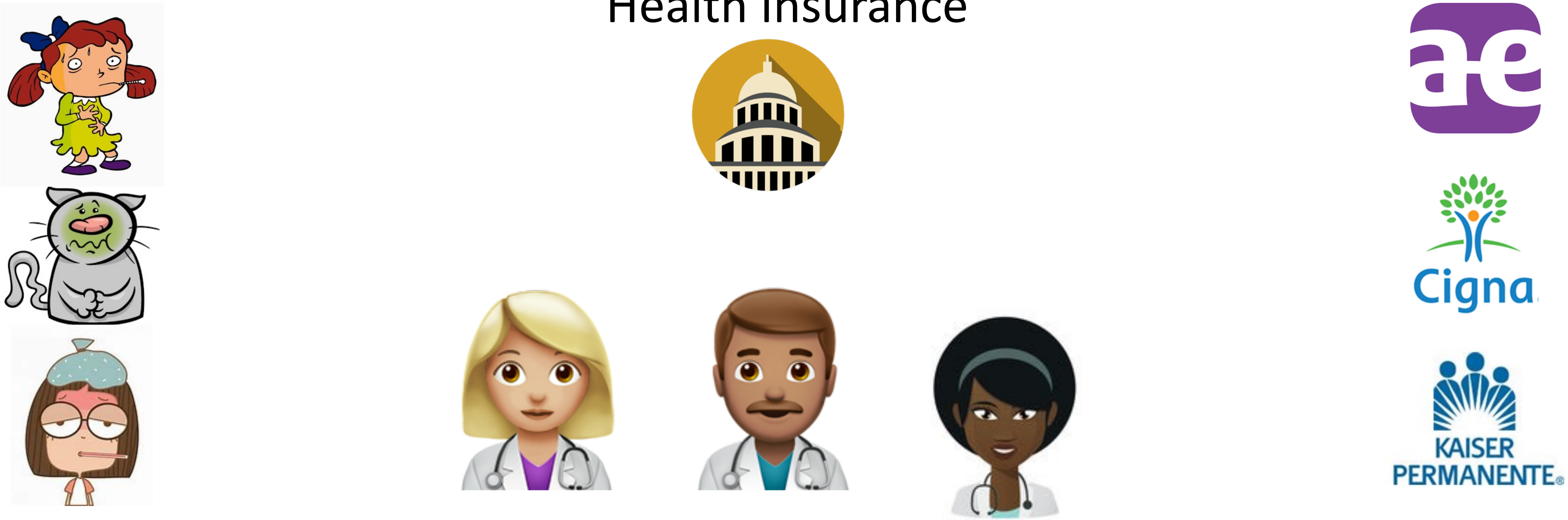
Settings where:

- Allocations are a mess.
- There are perverse incentives.

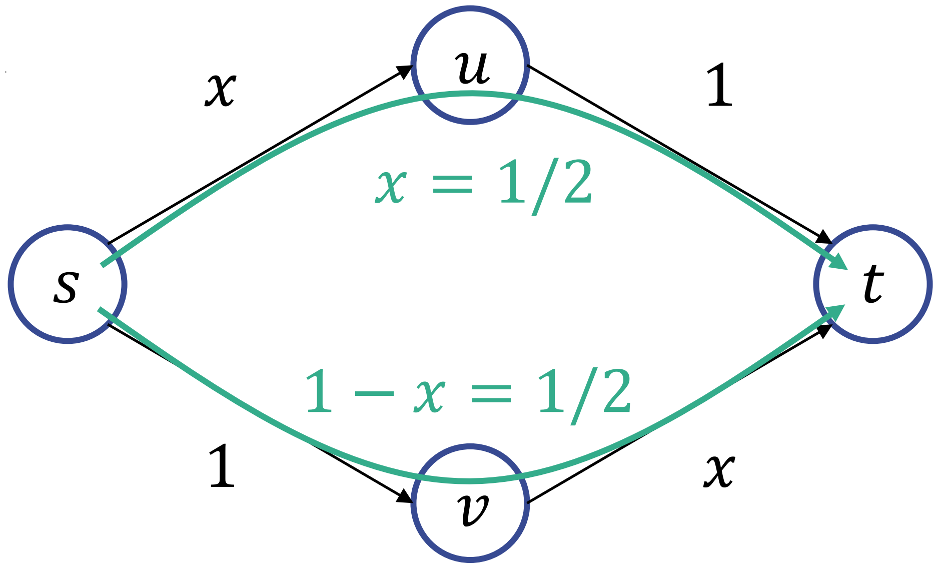
Computationally Efficient:

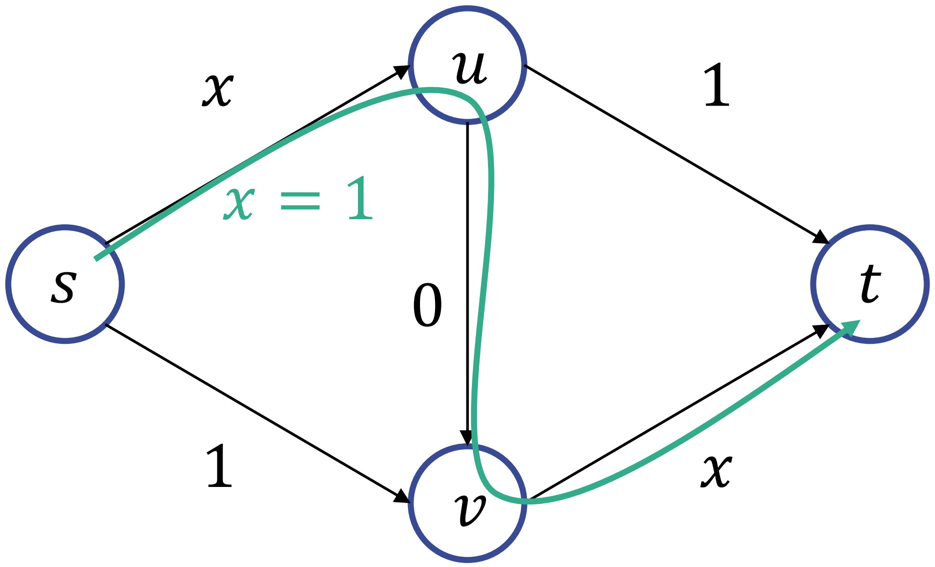
- To design.
- To run.
- To strategize within.

Health Insurance



Braess's Paradox





Fraction of population on route

Cost (think: time to travel with traffic)

Takeaways:
Adding a 0-cost road doesn't always help!
Agents don't choose what's best for them!

Centralized OPT

Price of Stability (PoS)

Price of Anarchy (PoA)

What should you expect to learn?

- Mechanism Design basics (welfare, revenue, environments)
    - Similar to other MD/EconCS courses. Probably the only part that is.
- LP Duality applied to mechanism design
- Robustness
- Mechanism Design for Social Good
- New frontiers (two-sided markets, contracts, dynamic MD)
- Data science & MD

Where can go you after this course?

Research in related fields:

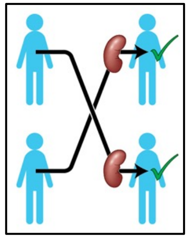
- EconCS (from CS)
- Operations Research (IE or Business)
- Microeconomic theory
- Some interdisciplinary split!

Add incentives or an economics perspective to your research:

- Privacy for strategic agents
- Learning with strategic agents

Related industries:

- Platform economics
- Allocation systems in welfare or industry
- Legal regulation (when is regulation better than markets?)



Let’s get started!