CPS 173: Computational Microeconomics

Instructor: Vincent Conitzer
(Assistant Professor of Computer Science & Economics)
conitzer@cs.duke.edu
http://www.cs.duke.edu/courses/spring10/cps173/

What is Economics?

- "the social science that studies the production, distribution, and consumption of goods and services." [Wikipedia, Jan. 2010]
- Some key concepts:
  - Economic agents or players (individuals, households, firms, bots, …)
  - Agents' current endowments of goods, money, skills, …
  - Possible outcomes ((re)allocations of resources, tasks, …)
  - Agents' preferences or utility functions over outcomes
  - Agents' beliefs (over other agents' utility functions, endowments, production possibilities, …)
  - Agents' possible decisions/actions
  - Mechanism that maps decisions/actions to outcomes

An economic picture

v( ) = 200

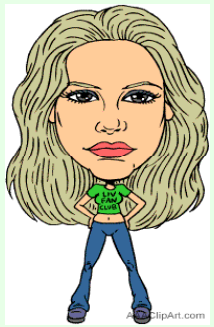
$ 800

v( ) = 100

v( ) = 400

v( ) = 200

v( ) = 400



$ 600

$ 200

After trade (a more efficient outcome)

v( ) = 200

$ 1100

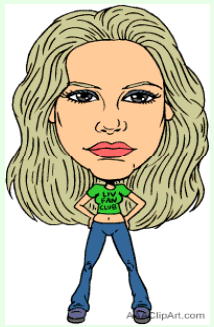
… but how do we get here? Unstructured trade? Auctions? Exchanges?

v( ) = 100

v( ) = 400

v( ) = 200

v( ) = 400



$ 400

$ 100

Some distinctions in economics

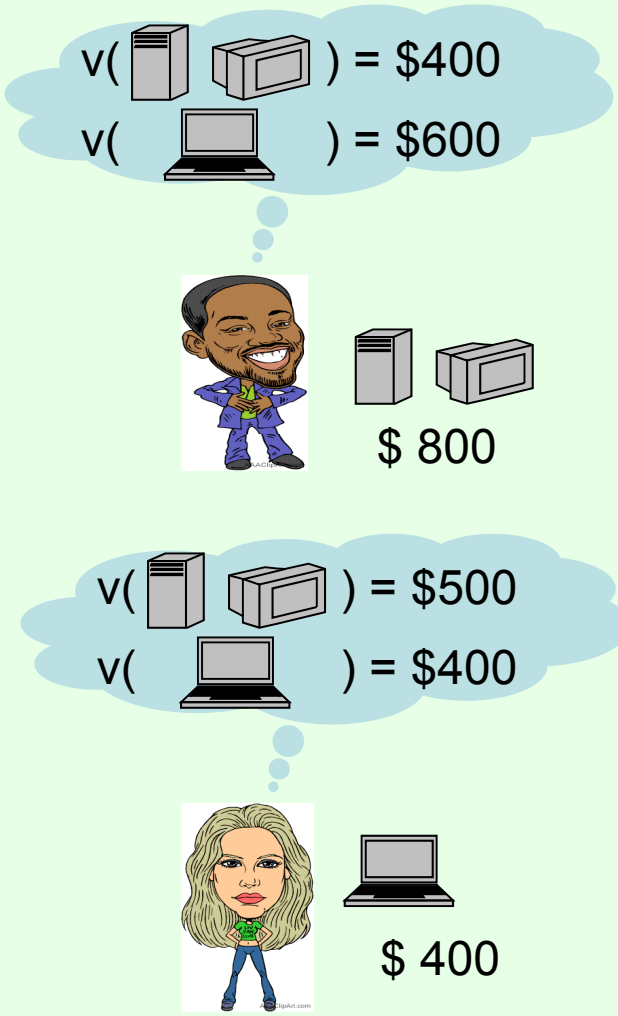
- Descriptive vs. normative economics
  - Descriptive:
    - seeks only to describe real-world economic phenomena
    - does not care if this is in any sense the "right" outcome
  - Normative:
    - studies how people "should" behave, what the "right" or "best" outcome is
- Microeconomics vs. macroeconomics
  - Microeconomics: analyzes decisions at the level of individual agents
    - deciding which goods to produce/consume, setting prices, …
    - "bottom-up" approach
  - Macroeconomics: analyzes "the sum" of economic activity
    - interest rates, inflation, growth, unemployment, government spending, taxation, …
    - "big picture"

What is Computer Science?

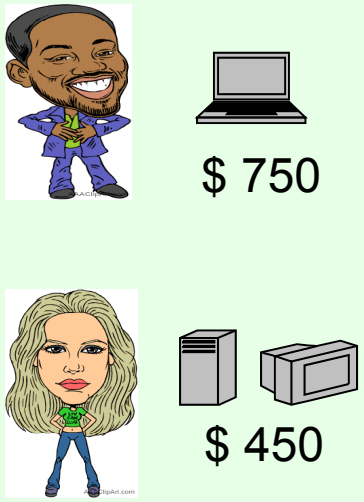
- "the study of the theoretical foundations of information and computation, and of practical techniques for their implementation and application in computer systems" [Wikipedia, Jan. 2010]
- A computational problem is given by a function f mapping inputs to outputs
  - For integer x, let f(x) = 0 if x is prime, 1 otherwise
  - For initial allocation of resources + agent preferences x, let f(x) be the (re)allocation that maximizes the sum of utilities
- An algorithm is a fully specified procedure for computing f
  - E.g., sieve of Eratosthenes
  - A correct algorithm always returns the right answer
  - An efficient algorithm returns the answer fast
- Computer science is also concerned with building larger artifacts out of these building blocks (e.g., personal computers, the Internet, the Web, search engines, spreadsheets, artificial intelligence, …)

Resource allocation as a computational problem *(Part 1 of the course)*

input

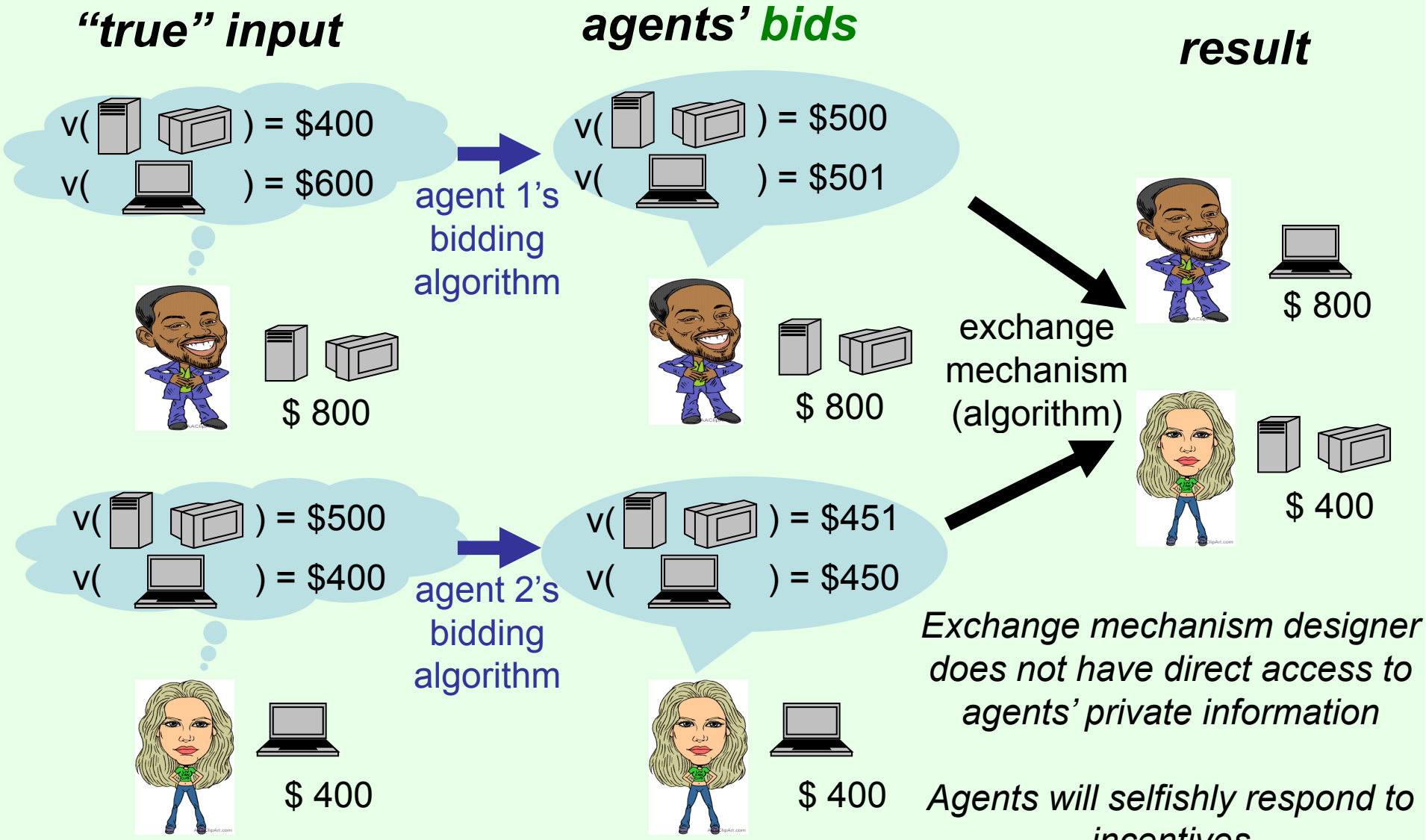


output



Here, gains from trade ($300) are divided evenly (not essential)

Economic mechanisms



Exchange mechanism designer does not have direct access to agents' private information

Agents will selfishly respond to incentives

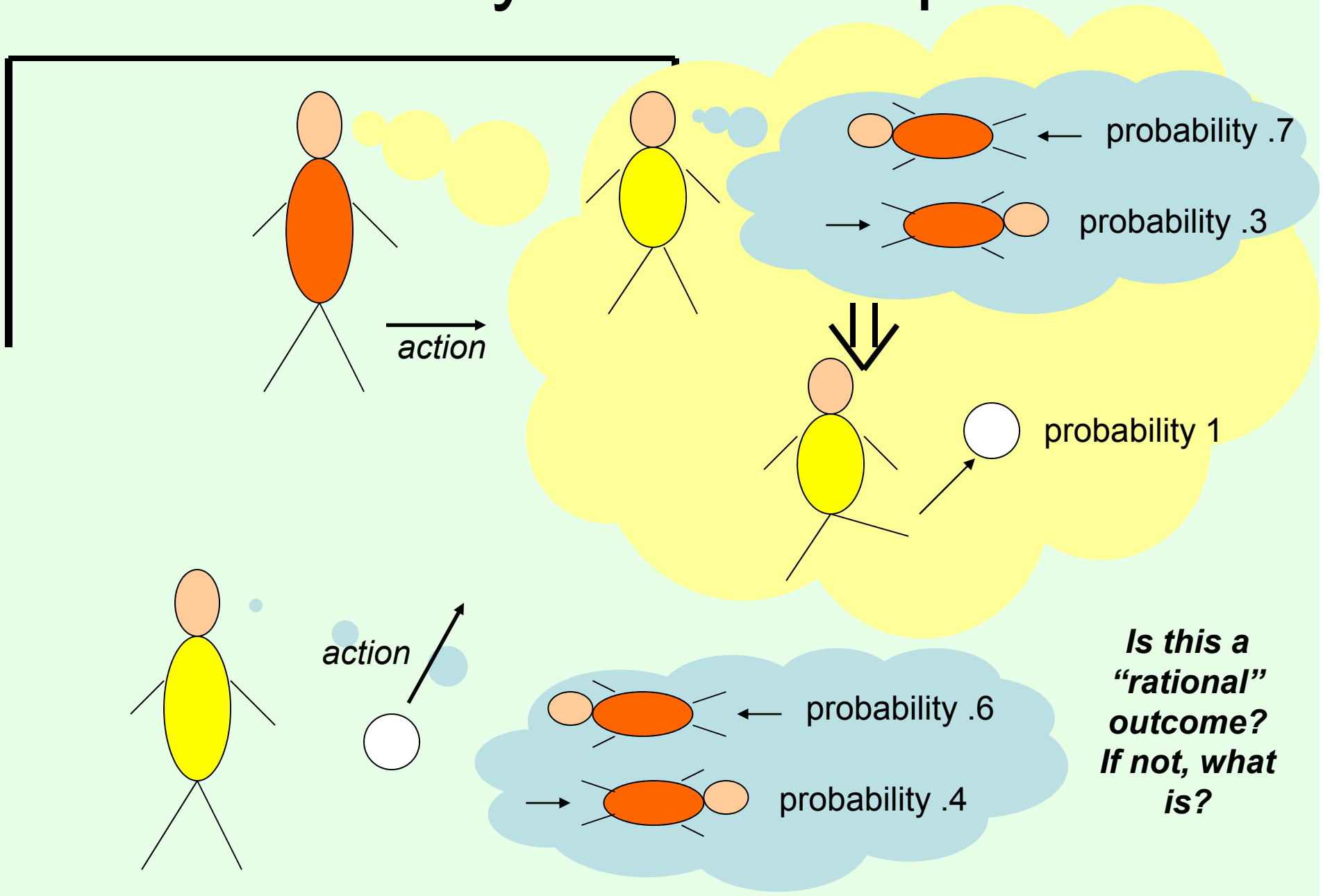
Game theory
(Part 2 of the course)

- Game theory studies settings where agents each have
  - different preferences (utility functions),
  - different actions that they can take
- Each agent's utility (potentially) depends on all agents' actions
  - What is optimal for one agent depends on what other agents do
    - Very circular!
- Game theory studies how agents can rationally form beliefs over what other agents will do, and (hence) how agents should act
  - Useful for acting as well as predicting behavior of others

What is game theory?

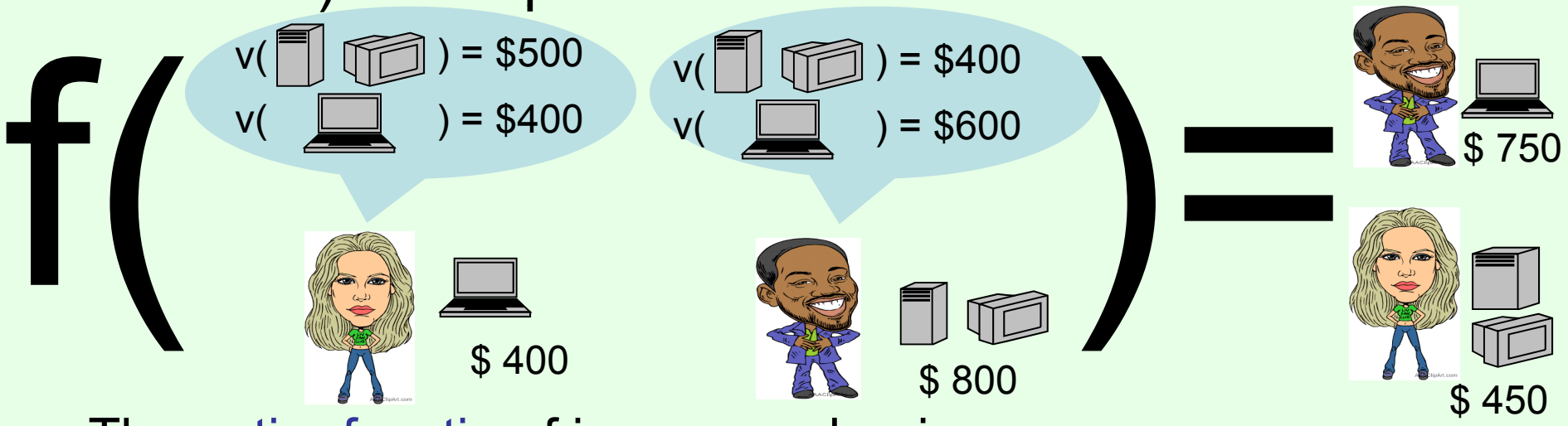
- “Game theory is a branch of applied mathematics that is used in the social sciences, most notably in economics, as well as in biology, engineering, political science, international relations, computer science, and philosophy. Game theory attempts to mathematically capture behavior in strategic situations, in which an individual's success in making choices depends on the choices of others.” [Wikipedia, Jan. 2010]

Penalty kick example



Mechanism design
(Part 3 of the course)

- Mechanism = rules of auction, exchange, …
- A function that takes reported preferences (bids) as input, and produces outcome (allocation, payments to be made) as output



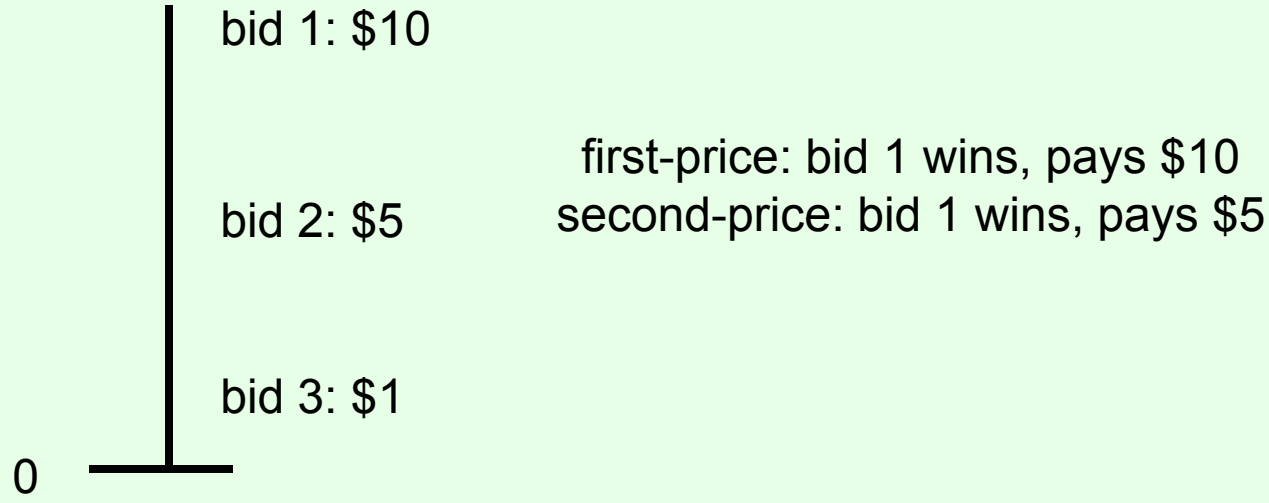
- The entire function f is one mechanism
- E.g., the mechanism from part 1: find allocation that maximizes (reported) utilities, distribute (reported) gains evenly
- Other mechanisms choose different allocations, payments

What is mechanism design?

- "[…] a field in game theory […] The distinguishing features [of mechanism design] are:
  - that a game "designer" chooses the game structure rather than inheriting one
  - that the designer is interested in the game's outcome
- […] usually solved by motivating agents to disclose their private information." [Wikipedia, Jan. 2010]

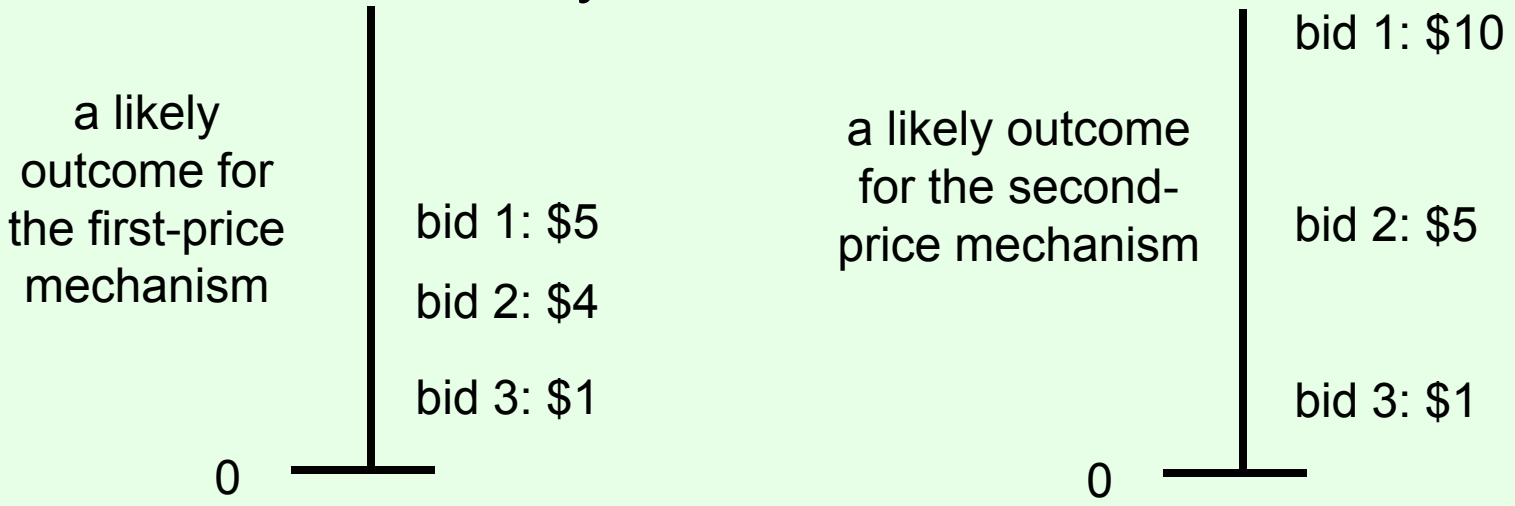
Example: (single-item) auctions

- Sealed-bid auction: every bidder submits bid in a sealed envelope
- First-price sealed-bid auction: highest bid wins, pays amount of own bid
- Second-price sealed-bid auction: highest bid wins, pays amount of second-highest bid



Which auction generates more revenue?

- Each bid depends on
  - bidder's true valuation for the item (utility = valuation - payment),
  - bidder's beliefs over what others will bid (→ game theory),
  - and... the auction mechanism used
- In a first-price auction, it does not make sense to bid your true valuation
  - Even if you win, your utility will be 0…
- In a second-price auction, (we will see later that) it always makes sense to bid your true valuation



Are there other auctions that perform better? How do we know when we have found the best one?

Mechanism design…

- Mechanism = game
- → we can use game theory to predict what will happen under a mechanism
  - if agents act strategically
- When is a mechanism "good"?
  - Should it result in outcomes that are good for the reported preferences, or for the true preferences?
  - Should agents ever end up lying about their preferences (in the game-theoretic solution)?
  - Should it always generate the best allocation?
  - Should agents ever burn money?(!?)
- Can we solve for the optimal mechanism?

How are we going to solve these problems? *(Part 0)*

- This is not a programming course

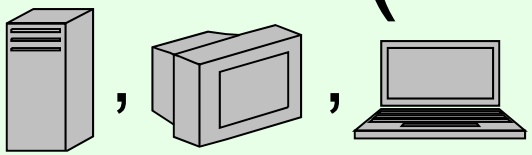
- Will use optimization software
  - GNU Linear Programming Kit (GLPK)
  - Linear programming, mixed integer linear programming

Uses of LP, MIP in this course

	Linear programming	**Mixed integer linear programming**
Part 1 (expressive marketplaces)	Winner determination in auctions, exchanges, … with partially acceptable bids	Winner determination in auctions, exchanges, … without partially acceptable bids
Part 2 (game theory)	Dominated strategies Minimax strategies Correlated equilibrium Optimal mixed strategies to commit to	Nash equilibrium
Part 3 (mechanism design)	Automatically designing optimal mechanisms that use randomization	Automatically designing optimal mechanisms that do not use randomization

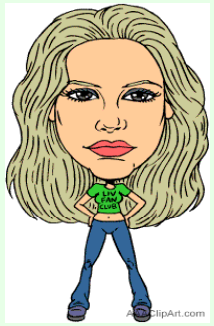
Other settings/applications

Combinatorial auctions (in Part 1)

Simultaneously for sale:  , ,

bid 1

v( ) = $500



bid 2

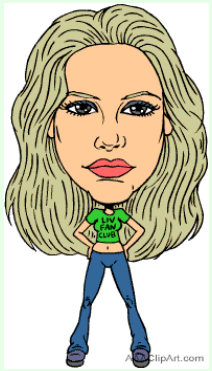
v( ) = $700

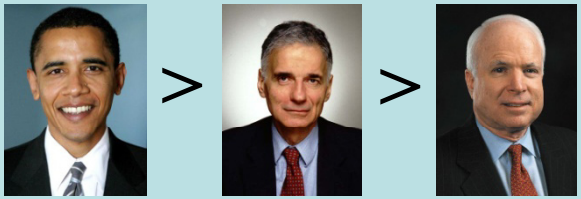
bid 3

v( ) = $300

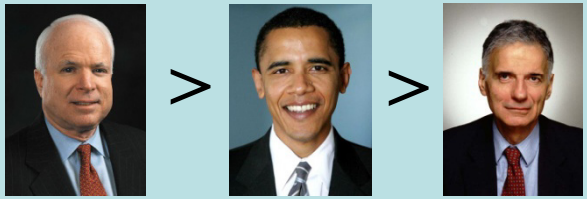
used in truckload transportation, industrial procurement, radio spectrum allocation, …

Voting (in Part 1)



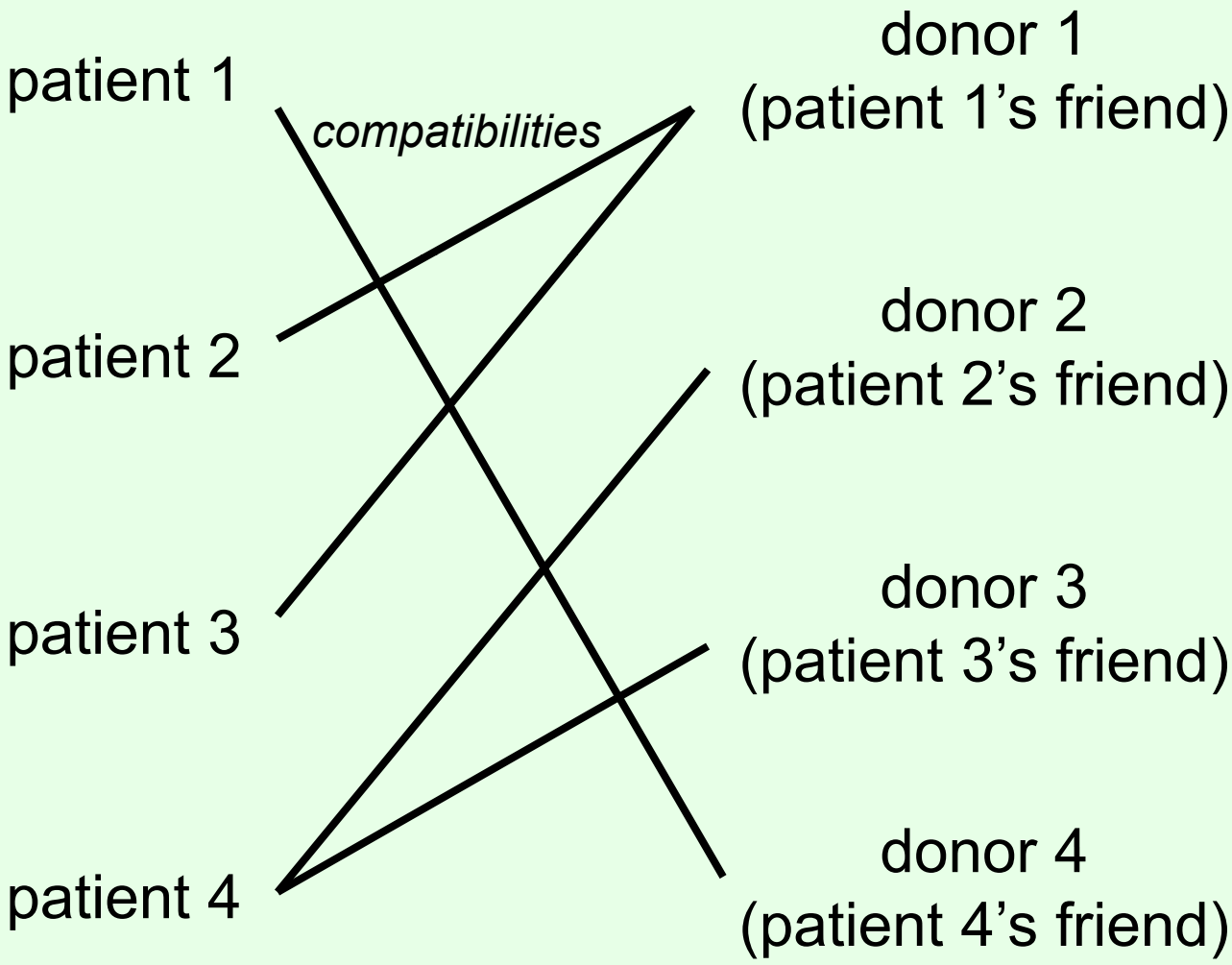


voting rule (mechanism) determines winner based on votes



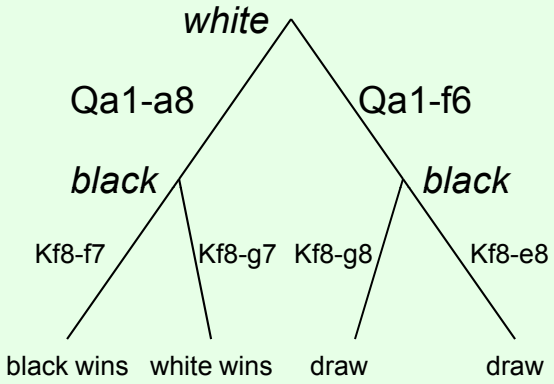
- Can vote over other things too
  - Where to go for dinner tonight, other joint plans, …
- Many different rules exist for selecting the winner

Kidney exchange (Part 1)



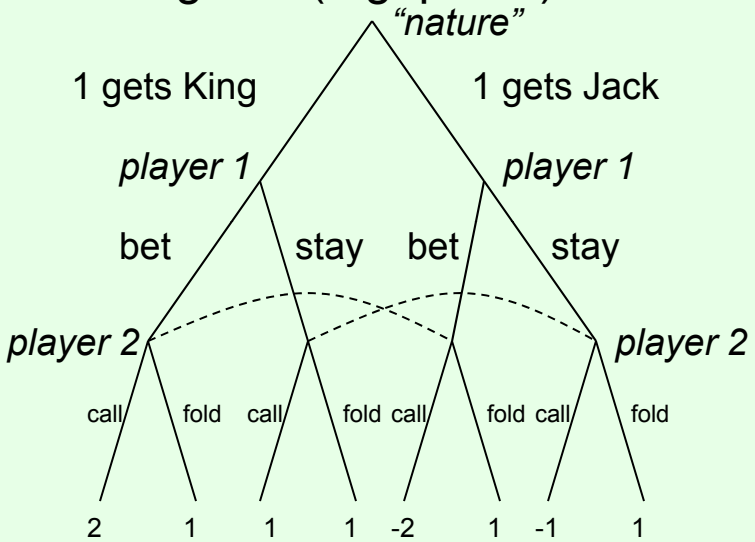
Game playing & AI (in Part 2)

perfect information games: no uncertainty about the state of the game (e.g. tic-tac-toe, chess, Go)



- Optimal play: value of each node = value of optimal child for current player (backward induction, minimax)
- For chess and Go, tree is too large
  - Use other techniques (heuristics, limited-depth search, alpha-beta, …)
- Top computer programs (arguably) better than humans in chess, not yet in Go

imperfect information games: uncertainty about the state of the game (e.g. poker)



- Player 2 cannot distinguish nodes connected by dotted lines
  - Backward induction fails; need more sophisticated game-theoretic techniques for optimal play
- Small poker variants can be solved optimally
- Humans still better than top computer programs at full-scale poker (at least most versions)
- Top computer (heads-up) poker players are based on techniques for game theory

Prediction markets

2010 US House of Representatives Control
The Democrats to control the House of Representatives after 2010 Congressional Elections

New: Request A Quote

SHARE

HOUSE.DEM.2010	Lifetime	Last Week	Last Day

Trade

Last Trade : 64.5 ▼ 0.5

Suggest An Image

Advanced graphing

Add this chart to your website/blog

Links
Related Intrade Markets
Contract Specific Rules
Search Yahoo News
Search Google

Contract Info	
Currency	USD
Session lo/hi	64.5 - 64.7
Life lo/hi	62.0 - 90.0
Previous Close	65.0
Open Price	64.5
Last Price	64.5
Last Trade Time	02:54 PM, GMT
Today's Volume	12
Total Volume	1281
Market Lifetime:	Sep 5 - Dec 31

BID		ASK	
Qty	Price	Price	Qty
12	64.5	64.8	3
15	64.4	65.8	2
100	64.3	65.9	1
1	64.1	66.8	2
2	63.5	67.8	4
1	63.2	68.5	22
8	63.0	69.0	4
6	62.6	69.6	1
80	62.0	69.7	2
1	61.5	69.8	10
130	60.0	69.9	5
4	55.0	70.0	20
1	50.0	71.9	5
1	1.0	73.3	1
		73.4	1

The Democrats to control the House of Representatives after 2010 Congressional Elections

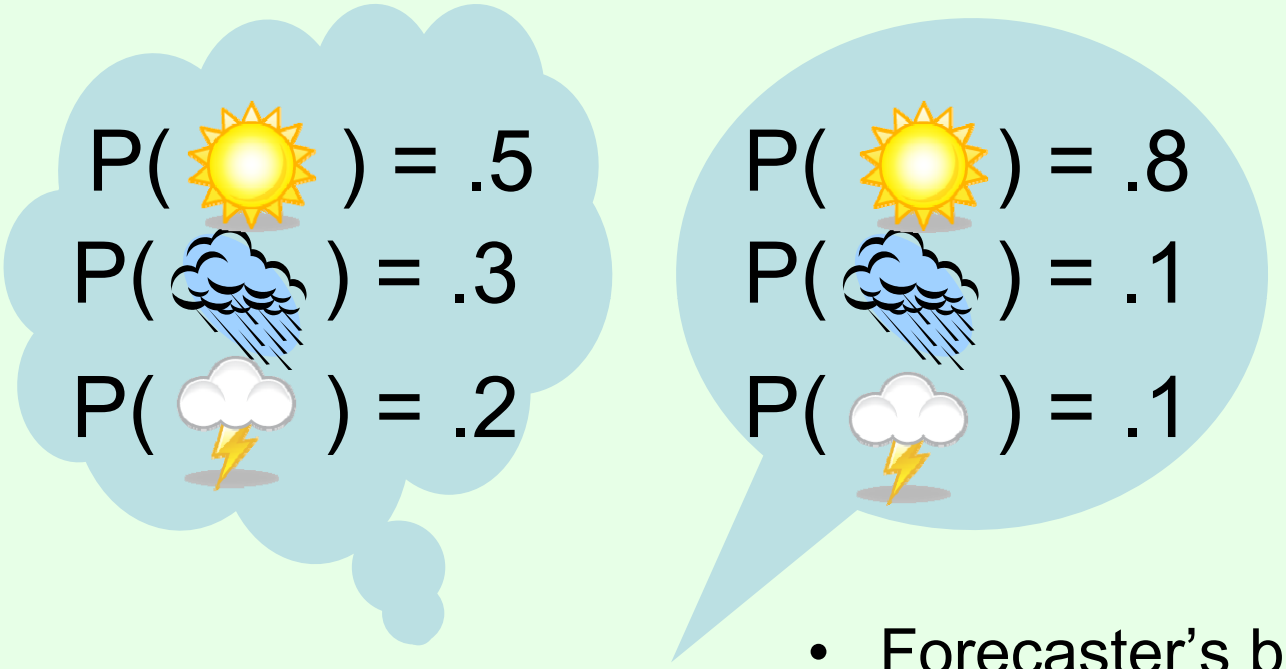
Contract Specs	Examples	Margin Specs	Explanation

Tick Size	0.1	Contract Start Date	Sep 5, 2008
Tick Value	USD 0.01	Max Contract Price	100.0

Financial securities (in Part 1)

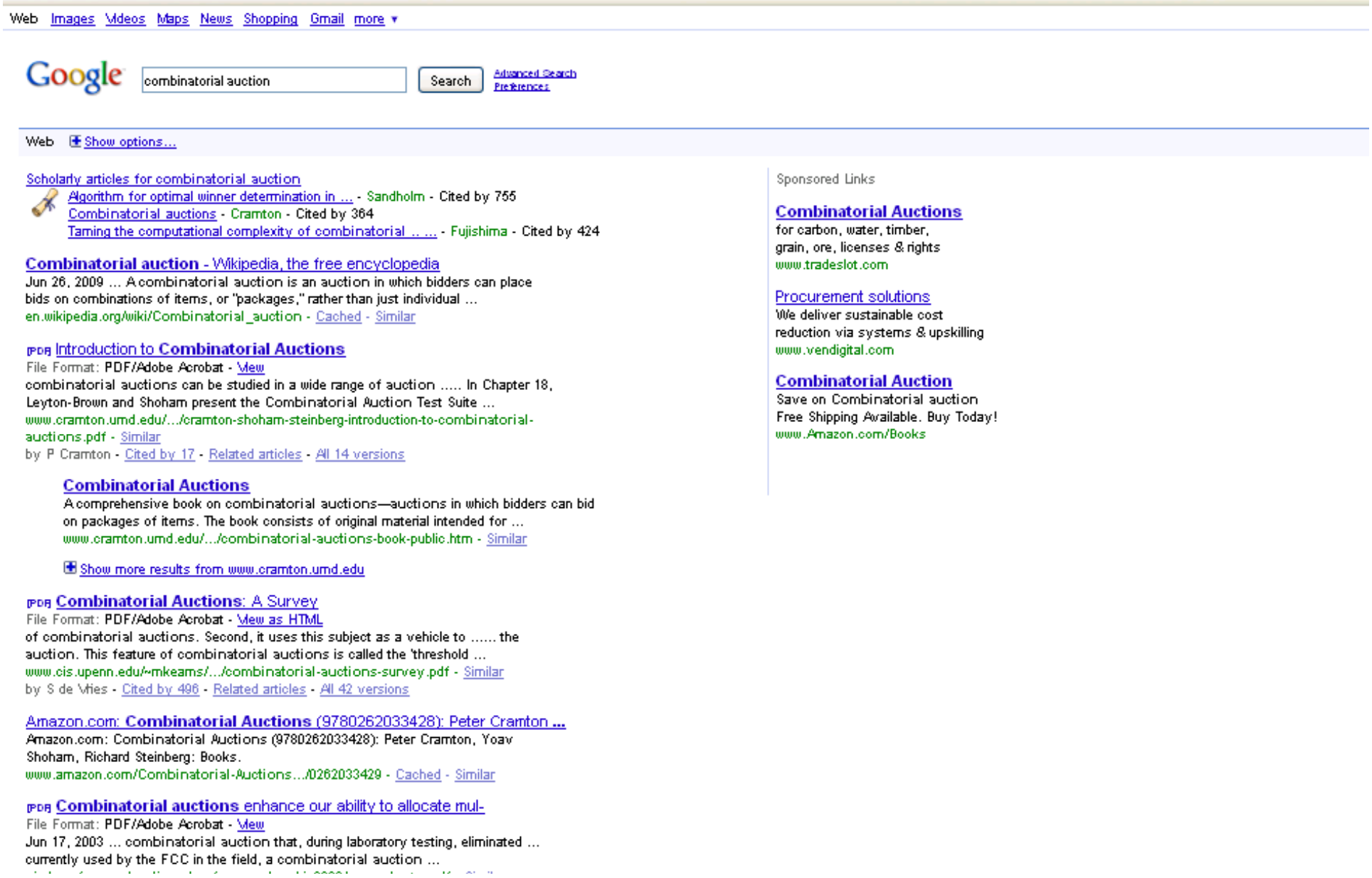
- Tomorrow there must be one of 
- Agent 1 offers \$5 for a security that pays off \$10 if  or 
- Agent 2 offers \$8 for a security that pays off \$10 if  or 
- Agent 3 offers \$6 for a security that pays off \$10 if 
- Can we accept some of these at offers at no risk?

How to incentivize a weather forecaster (in Part 3)



- Forecaster's bonus can depend on
  - Prediction
  - Actual weather on predicted day
- Reporting true beliefs should maximize expected bonus

Sponsored search (in Part 3)



Why should economists care about computer science?

- Finding efficient allocations of resources is a (typically hard) computational problem
  - Sometimes beyond current computational techniques
  - If so, unlikely that any market mechanism will produce the efficient allocation (even without incentives issues)
  - Market mechanisms must be designed with computational limitations in mind
  - New algorithms allow new market mechanisms

Why should economists care about computer science…

- Agents also face difficult computational problems in participating in the market
  - Especially acting in a game-theoretically optimal way is often computationally hard
  - Game-theoretic predictions will not come true if they cannot be computed
    - Sometimes bad (e.g., want agents to find right bundle to trade)
    - Sometimes good (e.g., do not want agents to manipulate system)

Why should computer scientists care about economics?

- Economics provides high-value computational problems
- Interesting technical twist: no direct access to true input, must incentivize agents to reveal true input
- Conversely: Computer systems are increasingly used by multiple parties with different preferences (e.g., Internet)
- Economic techniques must be used to
  - predict what will happen in such systems,
  - design the systems so that they will work well
- Game theory is relevant for artificial intelligence
  - E.g., computer poker