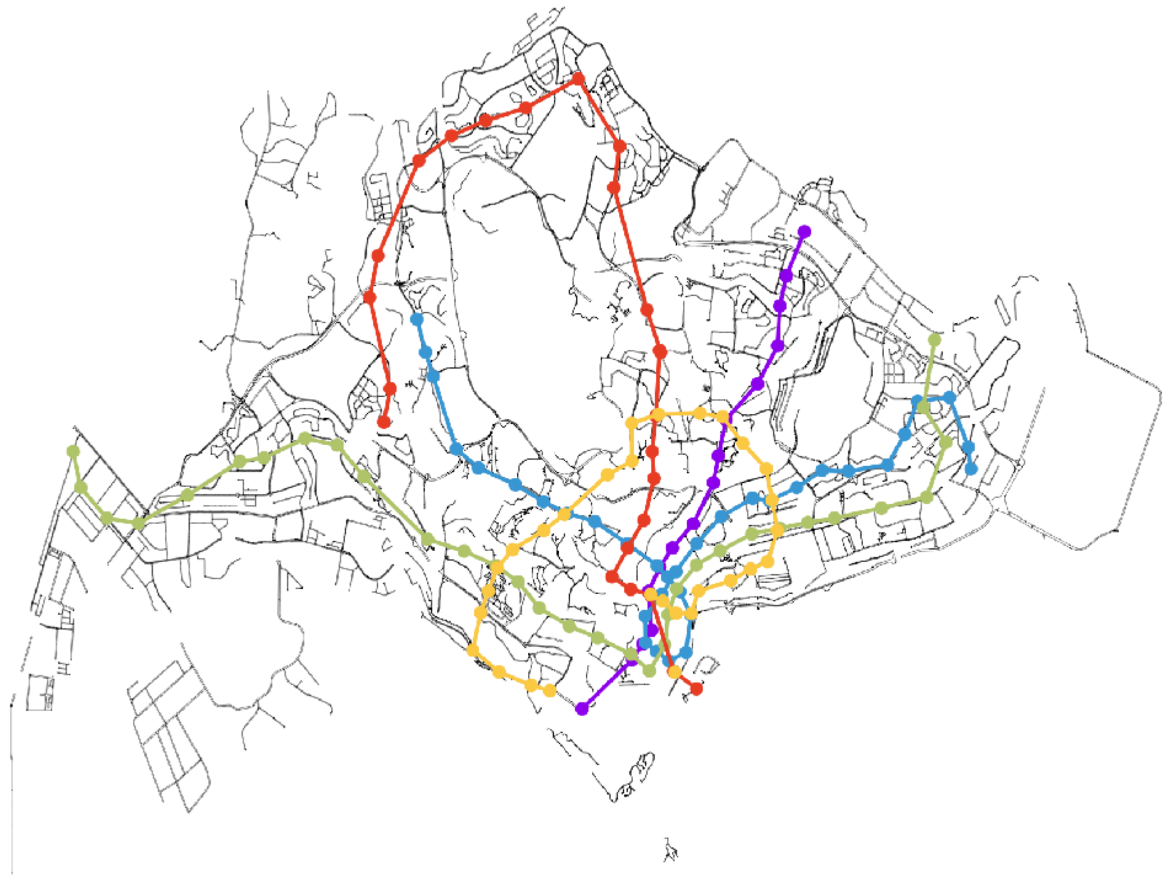
CSCI 357: Algorithmic Game Theory

Lecture 12: One-Sided Matching (without Money)

Shikha Singh



Announcements and Logistics

- No HW due this week 🎉
- Only 4 days to Spring Break! 🎉
- HW 4 budget agent competition results are in!
  - Will announce at the end of class given we have time
- HW 6 will be on the topics: Lectures 11 -13
- Will be released this week but due after you return from break
  - Regular assignment length, single person
  - Due Thursday April 7
- TA feedback form: will post on Slack, please fill by end of the week

Questions?

Last Time

- Decentralized matching market:
  - n buyers, $m \geq n$ items
- Goal: Match buyers to items and find a price vector $(p_1, \ldots, p_m)$ s.t.:
  - Matching is **envy free**
  - Market is **cleared**: no item with positive price (any demand) remains unmatched
- These are matched market-clearing prices
- Competitive equilibrium: $(M, \mathbf{p})$
- We created an ascending price auction show a competitive eq exists
  - Proved an **invariant** that every item with non-zero price is always matched
  - Means when algorithm terminates we have market-clearing prices!

Today

- Wrap up matching markets with money:
    - Prove that the ascending price auction terminates
- Move on to mechanism design without money:
    - Start with matching markets without money
    - One-sided markets today

Spring Break 🎉

Week 6: Matching Markets w/o Money

Week 5: Matching Markets w Money

Week 4: Bayesian Analysis & General Mechanism Design

Week 3: Application : Sponsored Ad Markets

Week 2: DSIC Auctions

Week 1: Game Theory

Ascending-Price Algorithm

- Start with prices of all items $p_j = 0$, assume all valuations $v_{ji} \in \mathbb{Z}$
- **Step 1.** Check if there is a **buyer-perfect matching** in preferred item graph
- **Step 2.** Else, there must a constricted set:
    - There exists $S \subseteq \{1,\ldots,n\}$ such that $|S| > |N(S)|$
    - $N(S)$ are items that are **over-demanded**
    - If there are multiple such sets, choose the **minimal set** $N(S)$
        - Increase $p_j \leftarrow p_j + 1$ for all items in the set $j \in N(S)$
    - Go back to **Step 1.**
- **Invariant:** if an item has non-zero cost, that item is tentatively matched to some buyer: $p_j > 0 \implies \exists i : (j, i) \in M$

Ascending-Price Algorithm

- **Invariant:** if an item has non-zero cost, that item is tentatively matched to some buyer: $p_j > 0 \implies \exists i : (j, i) \in M$
- Final question:
  - Does this algorithm ever terminate?
- **Intuition:** Since items are always tentatively matched, prices cannot rise for forever, why?
  - At some point, no buyer would want the items!

Proving Our Algorithm Terminates

- **Theorem.** The ascending price auction terminates.
- **Proof.** Show that algorithm starts with a certain amount of **"potential energy"** which goes down by at least 1 in each iteration
- Let the potential of any round be defined as:

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

- where p_j is the price of item j in that round and u_i^* is the maximum utility i can obtain given prices $\mathbf{p}$ in that round

Proving Our Algorithm Terminates

- **Theorem.** The ascending price auction terminates.
- **Proof.**
- At the the beginning, all prices are zero and $u_i^* = \max_j v_{ij}$
- Thus, before the auctions starts $E_0 = \sum_i \max_j v_{ij}$
- To wrap up proof, we show
  - Potential can never be negative $E \geq 0$
  - Potential at each step goes down by at least 1
- Thus, in E_0 steps the algorithm terminates. ∎

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

Proving Our Algorithm Terminates

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

- **Lemma:** Potential energy E is always non-negative.
- **Proof.**
- If there is at least one item with price 0 then $u_j^* \geq 0$
    - Why is this true? Use our invariant!
    - Every non-zero priced item is matched, thus when $n-1$ items are matched, no need to raise the price of nth item
- Since prices are always are always nonnegative $E \geq 0$

Proving Our Algorithm Terminates

$$E = \sum_{\text{items } j} p_j + \sum_{\text{buyers } i} u_j^*$$

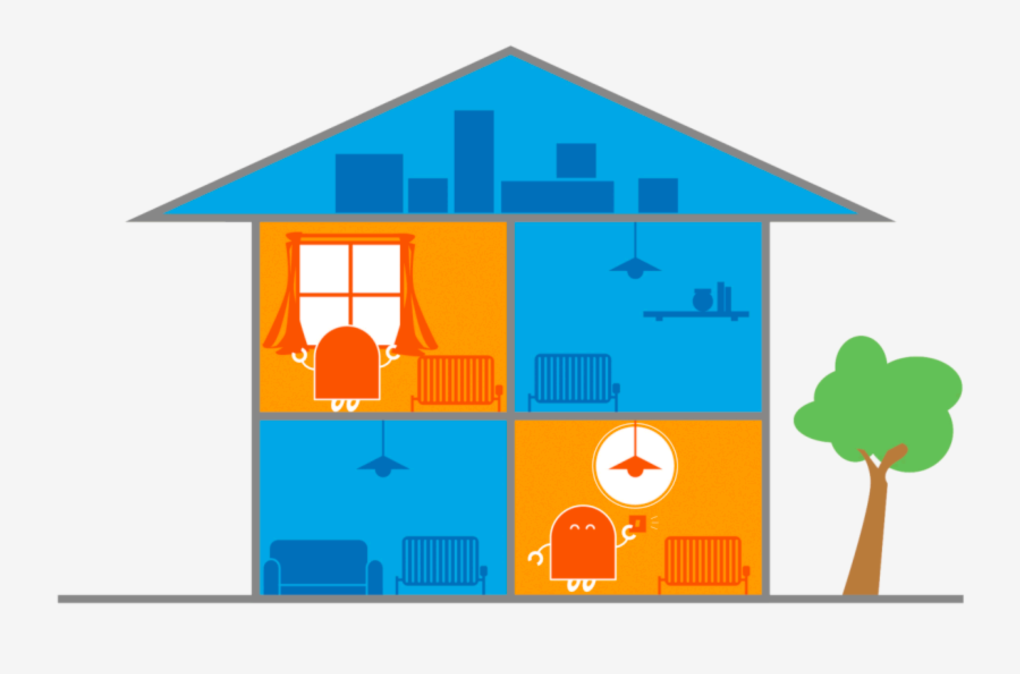
- **Claim.** Potential E goes down by at least one each step.
- **Proof**. At each step, we raise the price of all items in $N(S)$, how much does it increase the first term in E ?
  - $|N(S)|$
- However, the value of u_i^* goes down by one for each node in S, how much does this decrease the second term in E?
  - $|S|$
- Since $|N(S)| < |S|$, then potential decreases by at least 1
- Thus, the algorithm must terminate in E_0 steps ∎
- Our ascending auction terminates at market clearing prices!

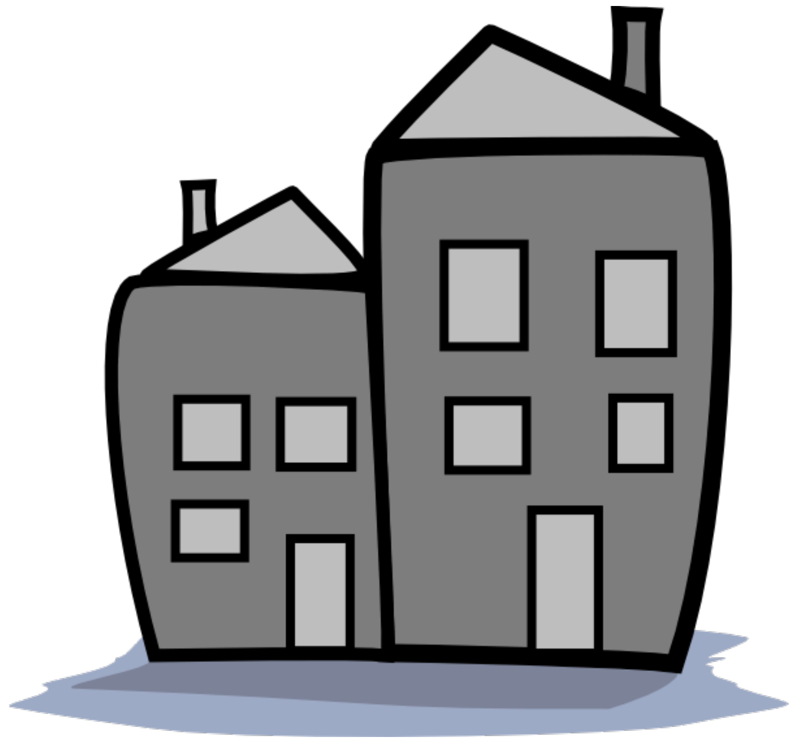
VCG Prices vs Market-Clearing

- VCG prices set **centrally**: ask each buyer to report their valuation and charge each buyer a **"personalized price"** for their allocation
- VCG prices are only set after a matching has been determined (the matching that maximizes total valuation of the buyers)
  - Not just about the item itself, but who gets the item
- Market-clearing prices are **"posted prices"** at which buyers are free to pick whatever item they like
  - prices are chosen first and posted on the item
  - Prices cause certain buyers to select certain items leading to a matching

Applying VCG

Prices

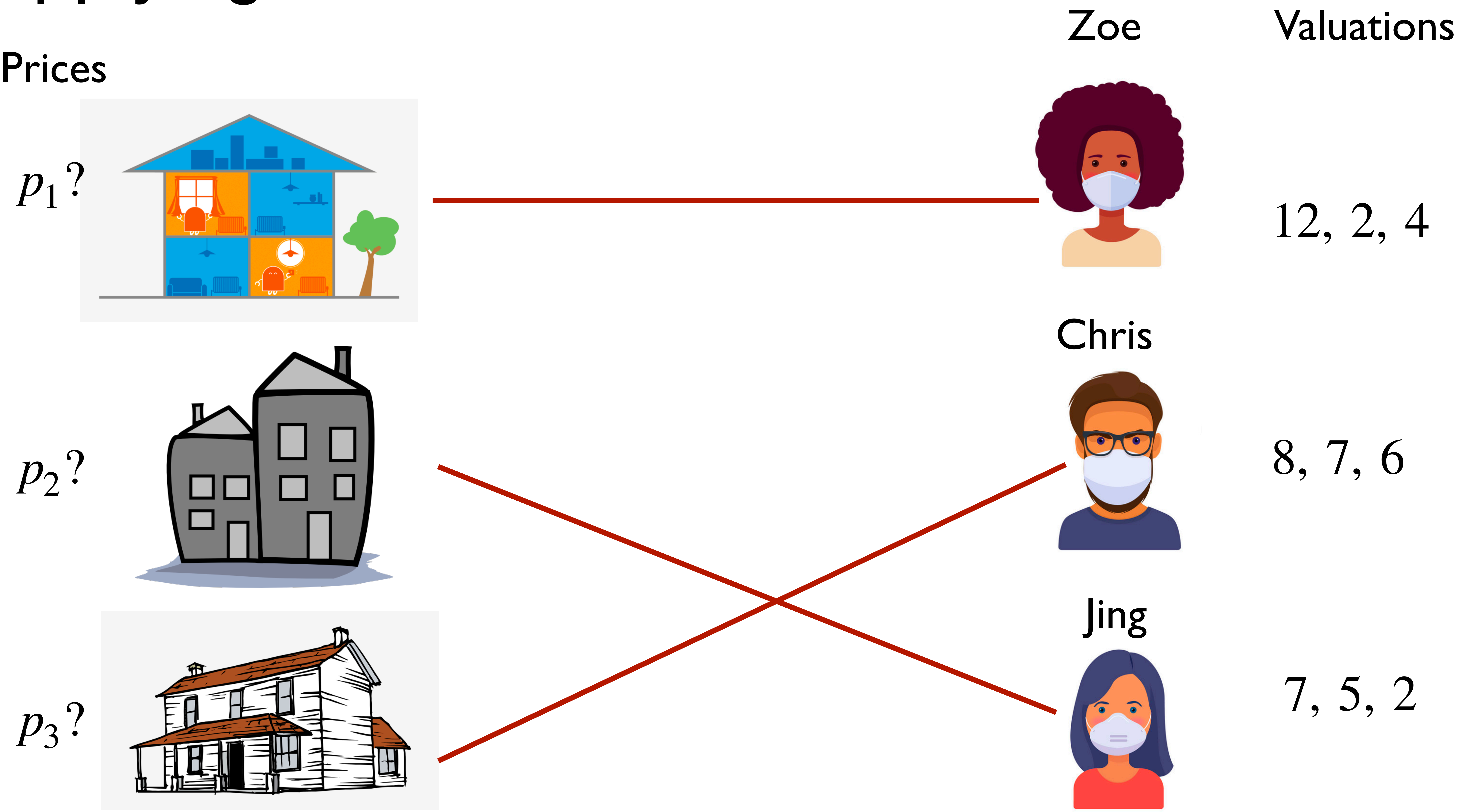




VCG. Need to find surplus maximizing allocation first

	Valuations
Zoe 	12, 2, 4
Chris 	8, 7, 6
Jing 	7, 5, 2

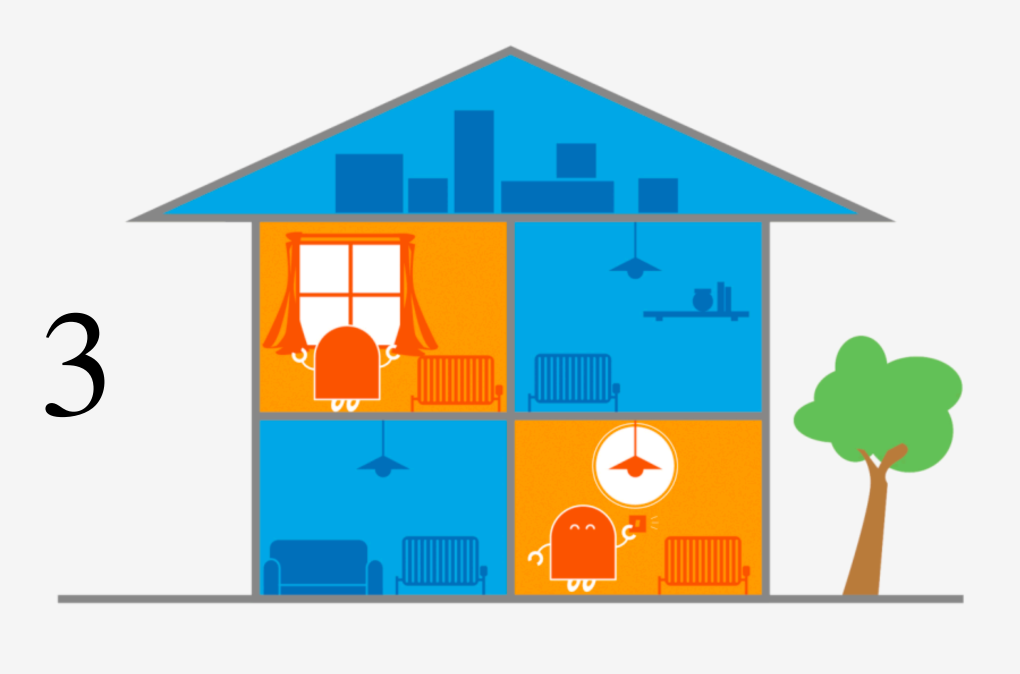
Applying VCG

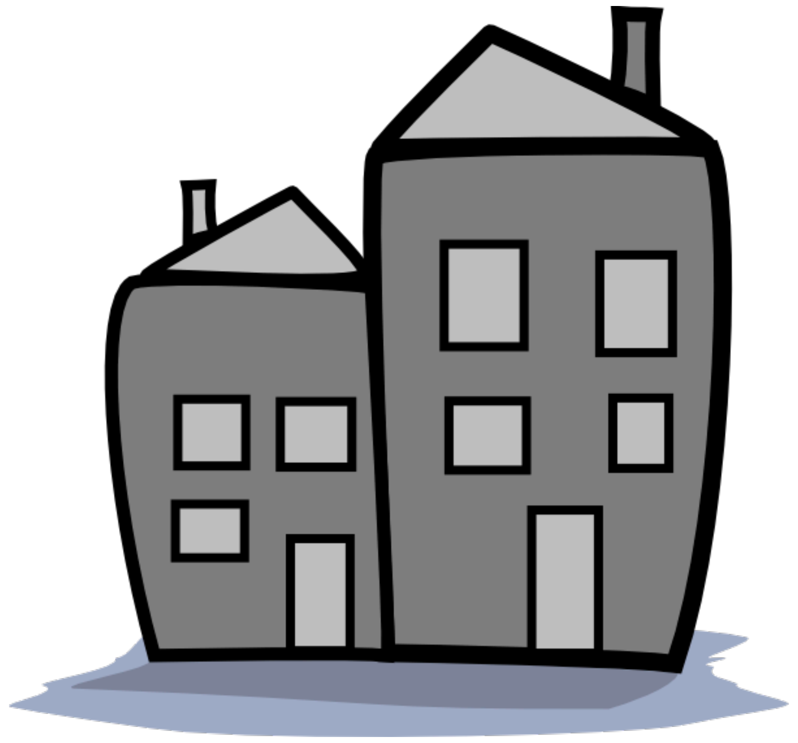


Applying VCG

Prices

$p_1 = 3$





Surplus without Zoe: **7+7 = 14**
Surplus by others when Zoe is present:
6 + 5 = 11

Zoe — Valuations

12, 2, 4

Chris

8, 7, 6

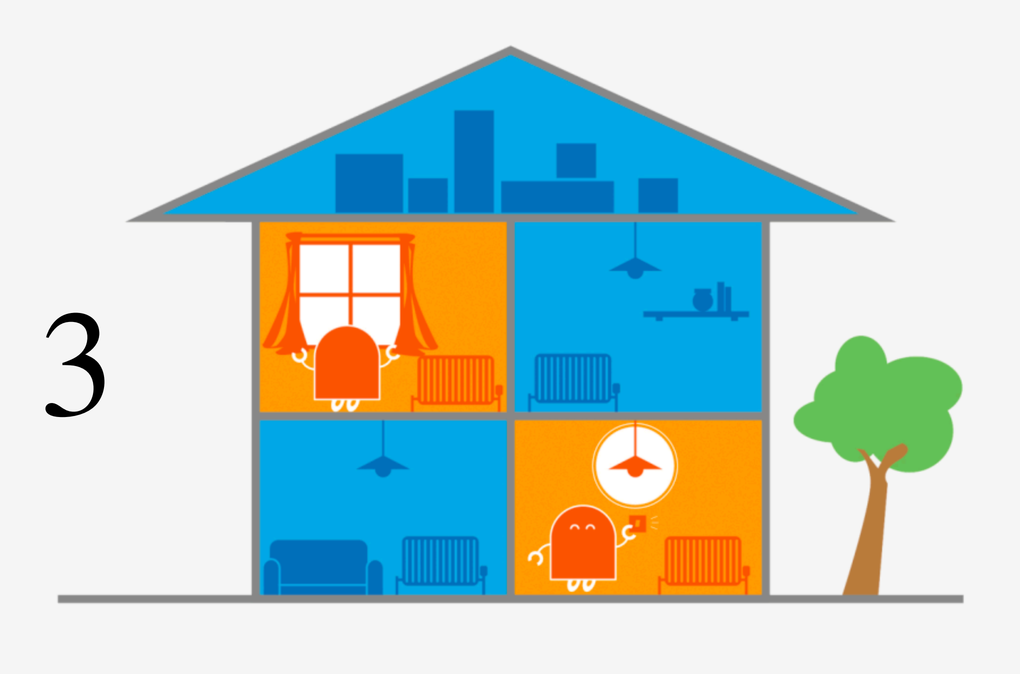
Jing

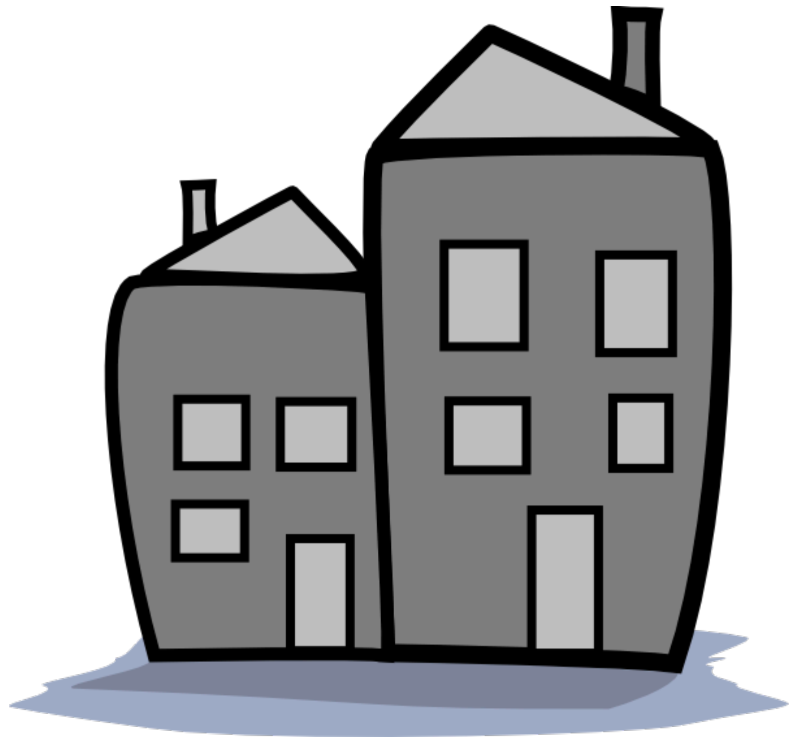
7, 5, 2

Applying VCG

Prices

$p_1 = 3$





$p_3 = 0$

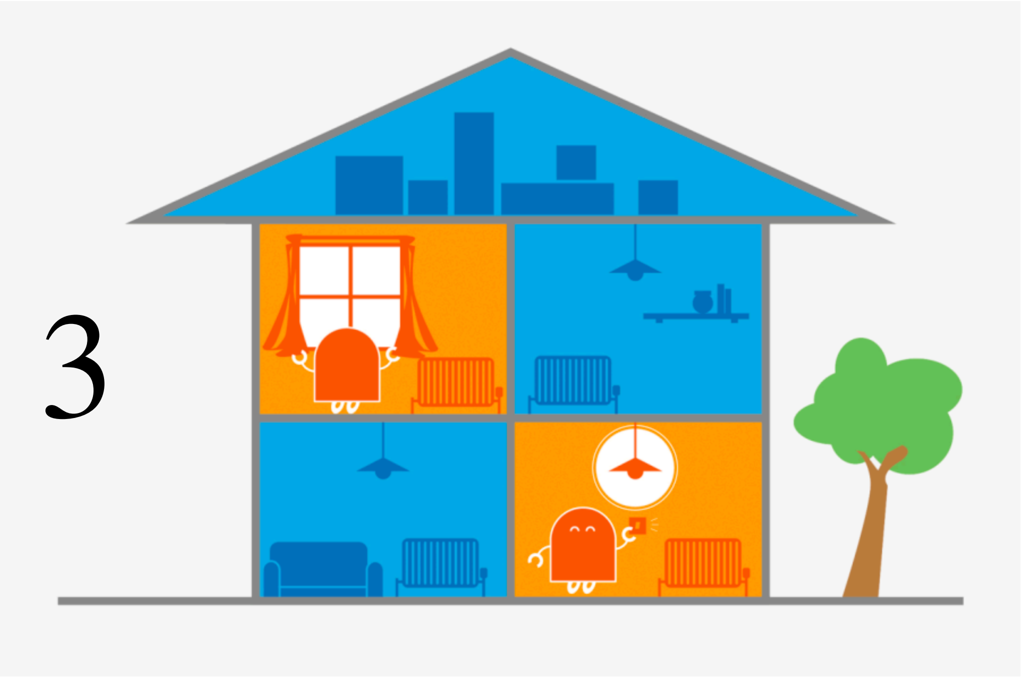
Surplus without Chris: **12+5 = 17**
Surplus by others when Chris is present: **12+5 = 17**

	Valuations
Zoe	12, 2, 4
Chris	8, 7, 6
Jing	7, 5, 2

Applying VCG

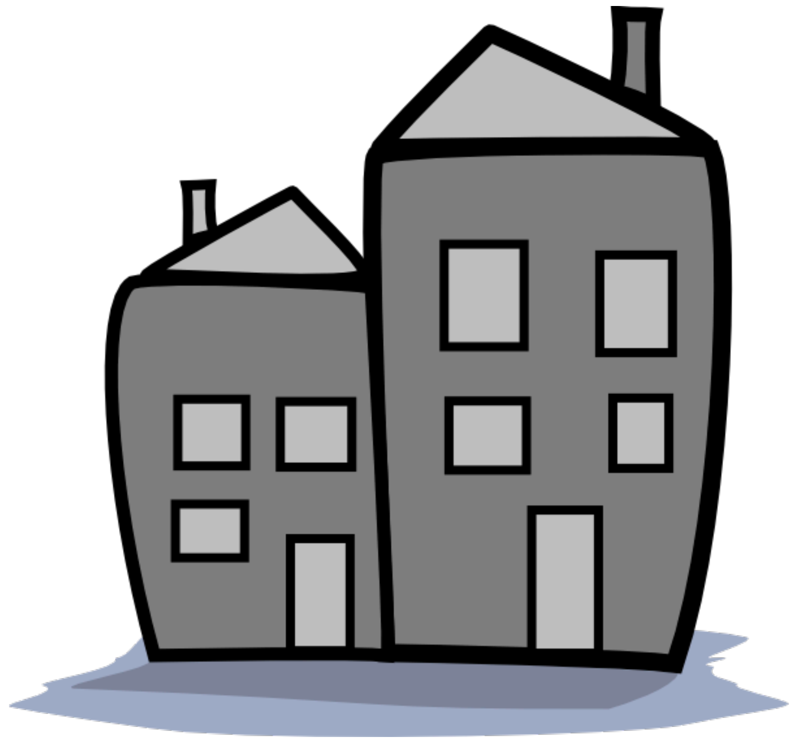
Prices

$p_1 = 3$



Surplus without Jing: **12+7 = 19**
Surplus by others when Jing is present:
12+6 = 18

$p_2 = 1$



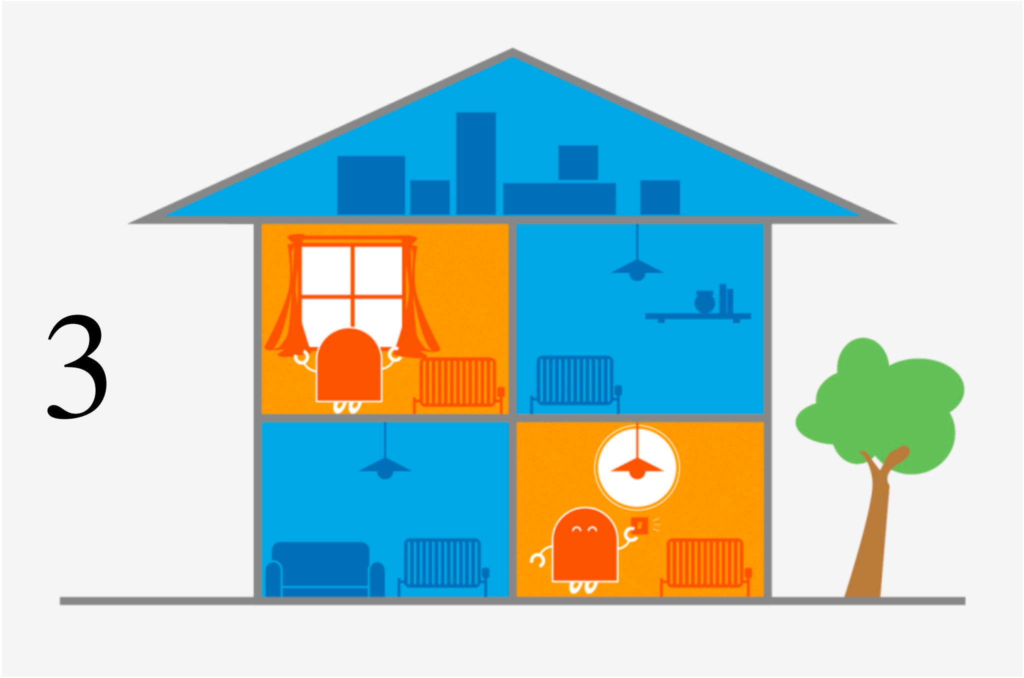
$p_3 = 0$

	Zoe	Valuations
		12, 2, 4
	Chris	
		8, 7, 6
	Jing	
		7, 5, 2

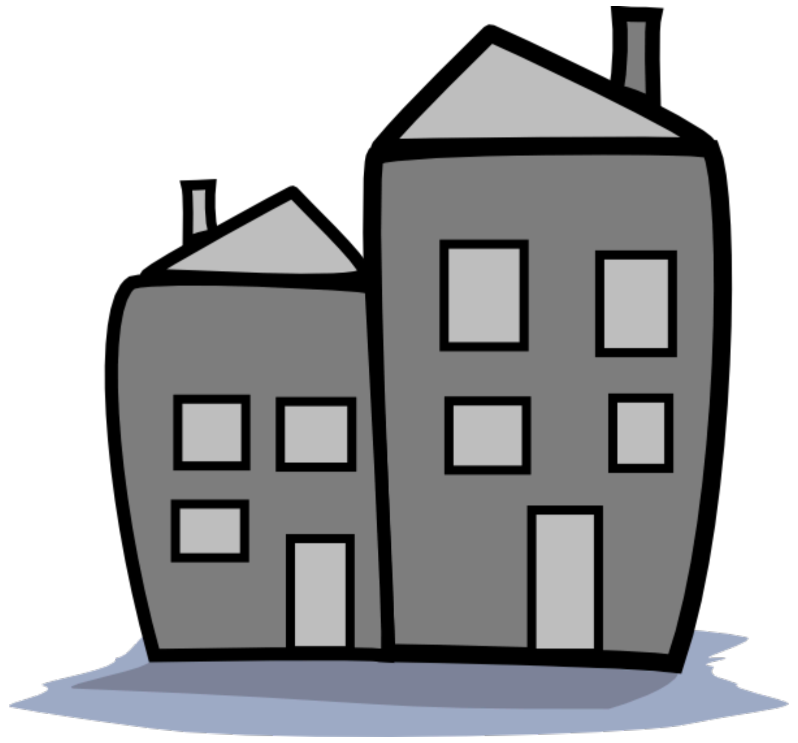
Applying VCG

Prices

$p_1 = 3$



$p_2 = 1$



$p_3 = 0$

We got the same **prices** & **matching** as our **competitive equilibrium**

	Valuations
Zoe 	12, 2, 4
Chris 	8, 7, 6
Jing 	7, 5, 2

VCG Prices are Market Clearing

- Despite their definition as personalized prices, VCG prices are always market clearing (for the case when each buyer wants a single item)
- Suppose we computed VCG prices for a given matching market
- Then, instead of assigning the VCG allocation and charging the price, we post the prices publicly
  - Without requiring buyers to follow the VCG match
- Despite this freedom, each buyer will in fact achieve the highest utility by selecting the item that was allocated by the VCG mechanism!
- **Theorem.** In any matching market (where each buyer can receive a single item) the VCG prices form the unique set of **market clearing prices of minimum total sum**.

This is a generalization of the VCG/GSP result (where valuations are constrained). The general proof is beyond the scope of this course.

General Demand

- Market clearing prices **may not exist in combinatorial markets**
- **Example**, suppose our market has two items $\{L, R\}$
- Two buyers Alice and Maya
- Alice wants both $v_a(\{L, R\}) = 5$, $v_a(\{L\}) = v_s(\{R\}) = 0$
- Maya wants either, $v_p(\{L\}) = v_p(\{R\}) = v_p(\{L, R\}) = 3$
- What's the welfare-maximizing allocation?
    - Give both to Alice
- What must the price of each be so that Maya doesn't want it?
    - $p(\{L\}) \geq 3, p(\{R\}) \geq 3$
- At a price of ≥ 6 does Alice want it?

Summary

- In a decentralized market with buyers and items, there exists a price $\mathbf{p}$ and matching M which form a competitive equilibrium
- Such an equilibrium can be reached by a simple **simultaneous ascending auction** that raises the price of "over-demanded" items
- Competitive equilibria are efficient: maximize social welfare and are guaranteed to exist
    - Does not extend to combinatorial demands but still useful in practice
- Caveats and direction of current research:
    - No sales occur until prices have settled at their equilibrium point
    - Coordination required for tie breaks

Competitive Equilibrium Research

- 2016 Article argues that competitive equilibrium's tie breaking requirement can be fairly strong
- Use **learning theory** to predict buyer's behavior and demand
- Show convergence under such some mild assumptions

Do Prices Coordinate Markets?

Justin Hsu*
Department of Computer and Information Science
University of Pennsylvania
USA
justhsu@cis.upenn.edu

Jamie Morgenstern†
Departments of Computer and Information Science and Economics
University of Pennsylvania
USA
jamiemmt@cs.cmu.edu

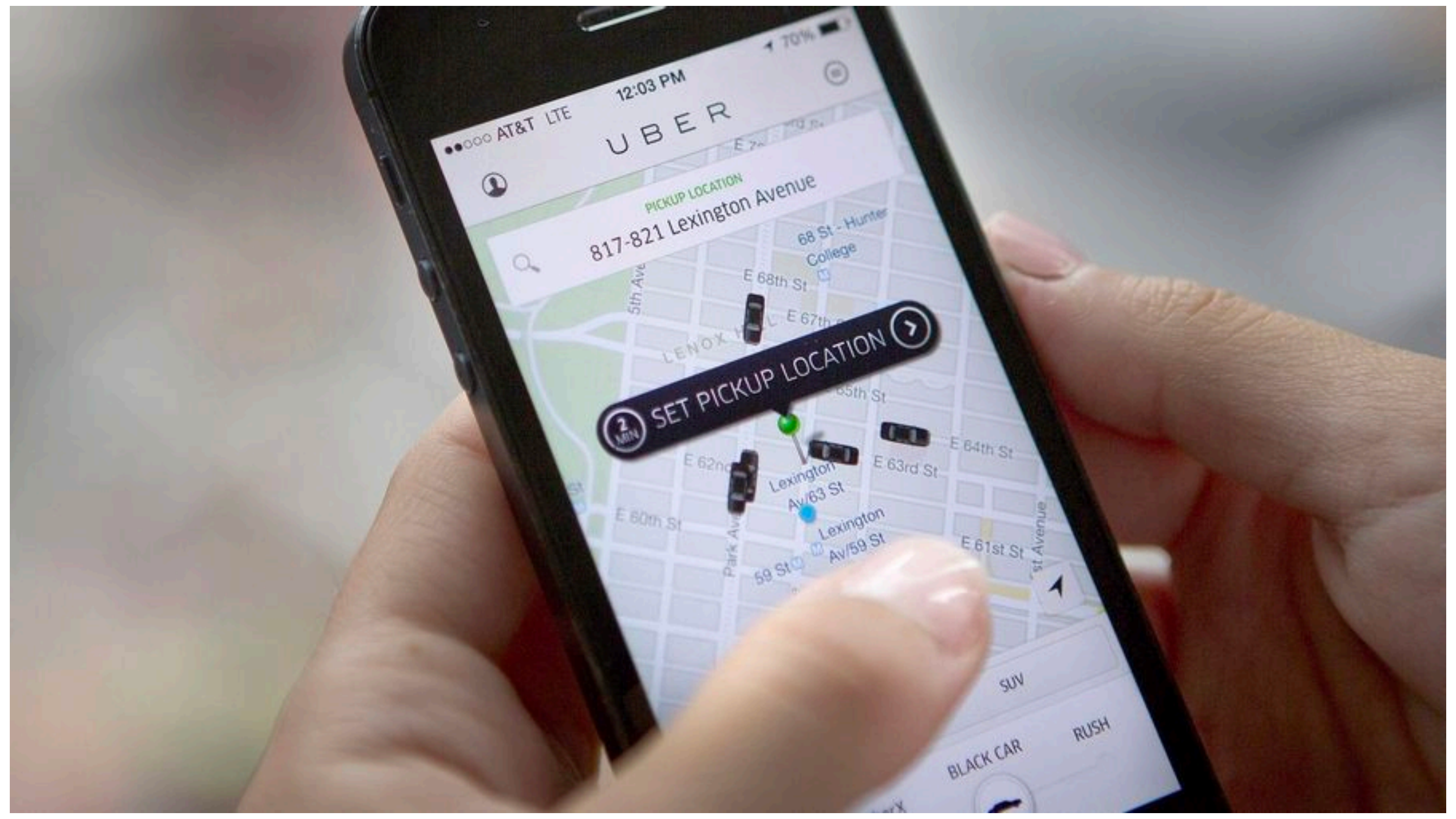
Ryan Rogers‡
Department of Applied Mathematics and Computational Science
University of Pennsylvania
USA
ryrogers@sas.upenn.edu

Aaron Roth§
Department of Computer and Information Sciences
University of Pennsylvania
USA
aaroth@cis.upenn.edu

Rakesh Vohra
Economics Department
University of Pennsylvania
USA
rvohra@seas.upenn.edu

Fluctuations in Practice: Research

- In practice, one might imagine that sales are actually happening concurrently with price adjustment
- It turns out, the way buyers and sellers respond to prices in the short-run can dramatically influence prices
- **Example**. Surge pricing on ride-sharing platforms can be viewed as an attempt to find market-clearing prices
- However, if passengers and drivers respond to prices myopically, the resulting behavior can be erratic
- Recent research in AGT studies **dynamic (online) resource allocation problems** that take these factors into account



Matching Markets (without Money)

Mechanism Design With Money

Designer's Goal: Allocate items to ensure good global guarantees (e.g. welfare)

Agent's Goal: Report **private preferences** so as to maximize their utility.

n agents with private preferences over items

Multiple items

Mechanism Design With Money

Designer's Goal: Allocate items to ensure good global guarantees (e.g. welfare)

Agent's Goal: Report **private preferences** so as to maximize their utility.

n agents with private preferences over items: expressed as **values** (cardinal)

Multiple items

Payments so far were a way to incentivize truthful behavior (strategyproof-ness)

Mechanism Design Without Money

Designer's Goal: Allocate items to ensure good **global guarantees**

Agent's Goal: Report **private preferences** that achieve **the best outcome**

n agents with private preferences over items (**ordinal**)

Multiple items

What are good global guarantees? How to incentivize truthful behavior without money?

Mechanism Design without Money

- Many domains money transfer is either infeasible or inappropriate or illegal
- Problem domains without money?
    - Matching students to courses
    - Matching students to school/ colleges/ dorms
    - Matching doctors to hospitals
- Sharing resources or barter markets:
    - Exchanging goods or services
- Social decision making:
    - Voting to elect a leader, a committee or an outcome

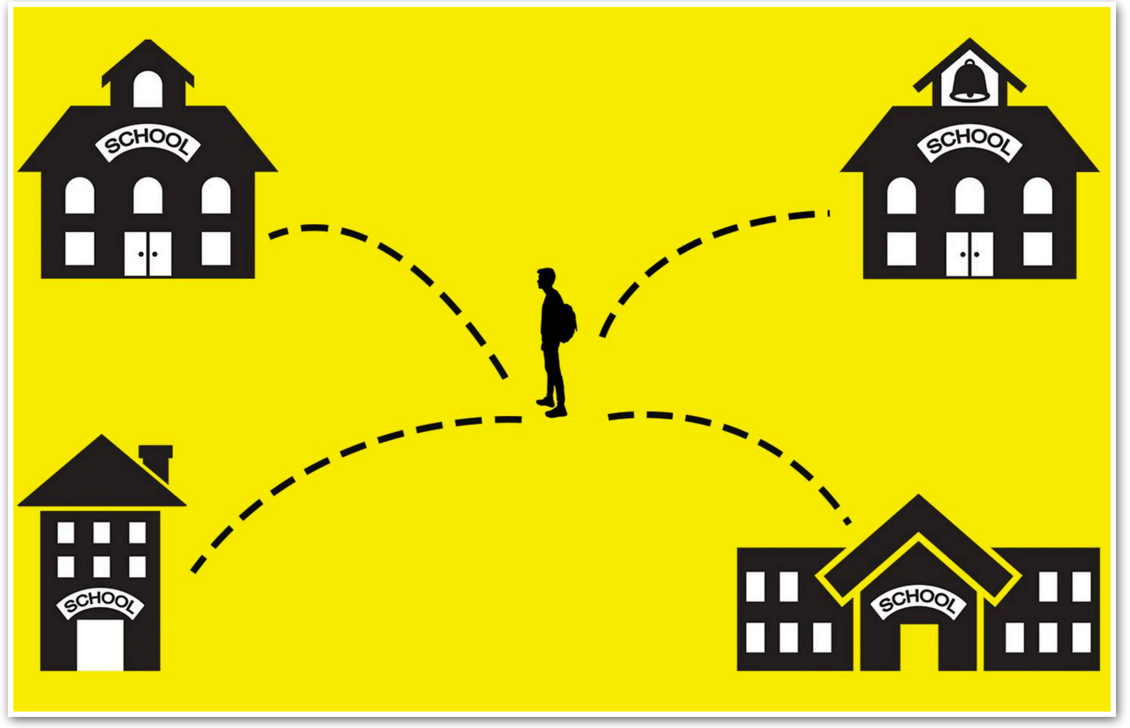
Domain of AGT where TCS truly shines!

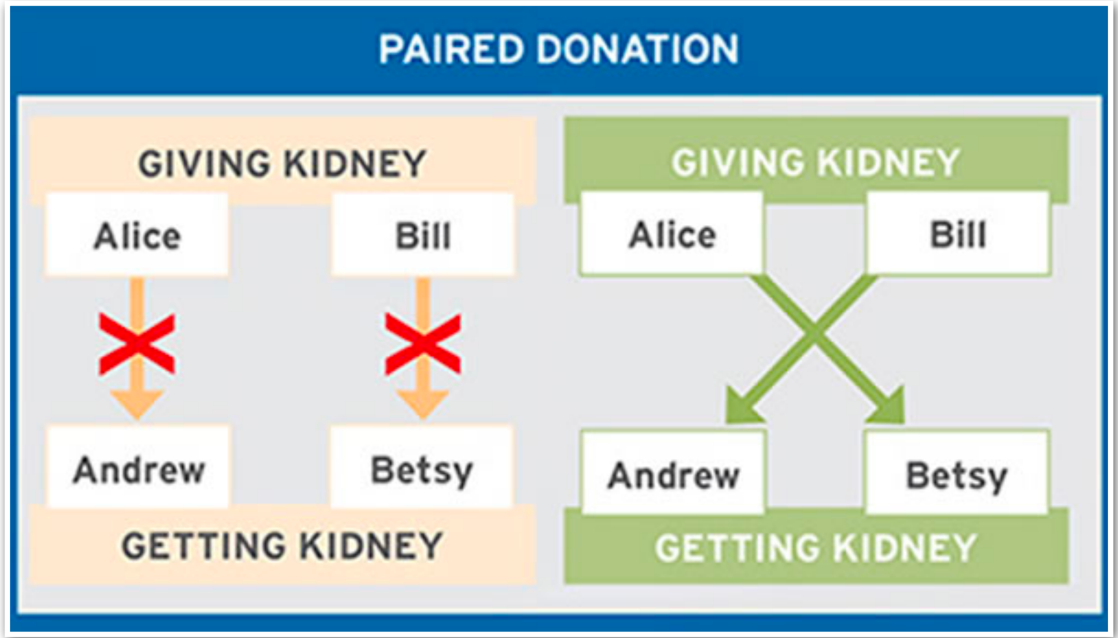
Matching Markets without Money

One Sided Markets

Two Sided Markets

No exchange





THE MATCH®
NATIONAL RESIDENT MATCHING PROGRAM®

Exchange based

One-Sided Matching

Designer's Goal: Allocate items to ensure good **global guarantees** ?

Agent's Goal: Report **private preferences** that achieve **the best outcome**

n students with **ordered preferences** over dorms

College Dorms

What are good global guarantees? How to incentivize truthful behavior without money?

One-Sided Matching

Designer's Goal: Allocate items to ensure **Pareto Optimality**

Agent's Goal: Report **private preferences** that achieve **the best outcome**

n students with **ordered preferences** over dorms

College Dorms

Pareto optimality: An outcome O is Pareto optimal if there is no outcome O' and where every agent does as well as in O and some agent does strictly better.

Assignment Problems

- One-sided matching problems: called allocation or assignment problems:
  - Assigning students to dorms
  - Offices to employees
  - Tasks to volunteers
- **Model**. We have n agents and n items
  - Agents have **strict preference ordering** over the items
    - Care only about their own allocation, not others
- Feasible assignment: matching between items and agents
- **Goal:**   Find a Pareto optimal assignment (means no other assignment can make an agent better off without making another agent worse off)

One-Sided Matching

Designer's Goal: Assignment of items to agents is **Pareto optimal**

Agent's Goal: Report **private preferences** that achieve **the best outcome**

n students with **ordered preferences** over dorms

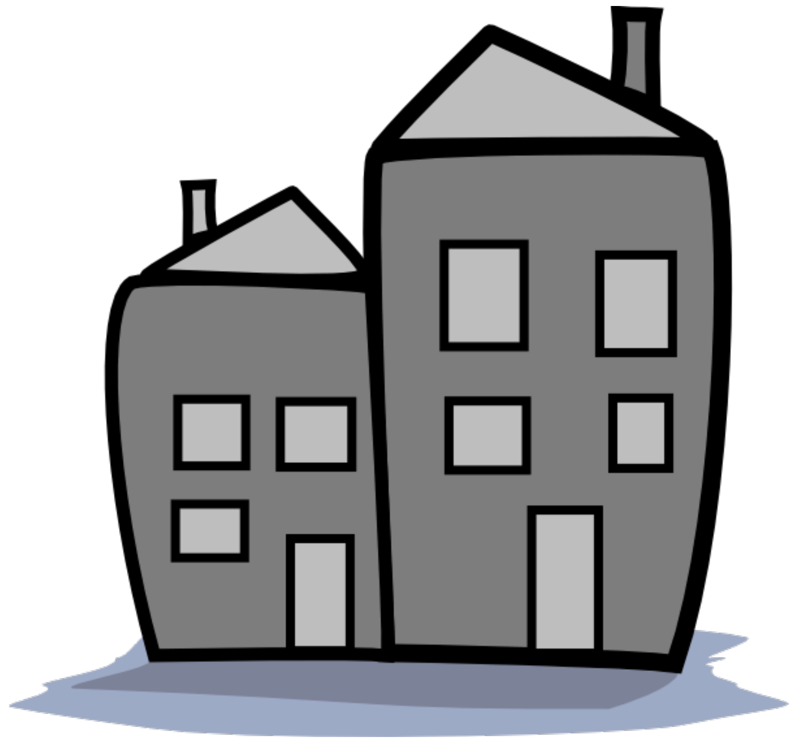
College Dorms

Mechanism. Any ideas for algorithms that incentivize truthful behavior?

One-Sided Matching Market

1

2



3

How do we matching students to dorms?

Private Preferences

Zoe

1 > 3 > 2

Chris

1 > 2 > 3

Jing

1 > 2 > 3

Housing Lotteries

- Most housing allocation algorithms look something like this:
  - Asks agents to report their preferences over items
  - Choose an ordering of all agents (lottery order)
    - Often based on some metrics are considered "fair", e.g., seniority, years of service to college, family size, etc
- Go down the list, assign each agent their **favorite item that is still remaining**
- **Example.** Faculty housing lottery at Williams
- This is a good mechanism?
  - Strategyproof, Pareto optimal?

Serial Dictatorships (SD)

- Each of the n agents submit a ranked ordering over items
- Each agent is assigned a rank from $\{1,2,\ldots,n\}$
- For $i = 1,2,\ldots,n$
  - Agent i is assigned their favorite choice among options still available
- **Lemma**. The serial dictatorship mechanism is strategyproof & Pareto optimal.
- Why is it strategyproof, that is, why is truthful reporting of preferences a dominant strategy for the agents
  - Cannot control lottery order
  - Given lottery order, truthful reporting obtains the best possible outcome
  - No incentive to deviate (regardless of other's preferences)

Serial Dictatorships (SD)

- Each of the n agents submit a ranked ordering over items
- Each agent is assigned a rank from $\{1,2,\ldots,n\}$
- For $i = 1,2,\ldots,n$
    - Agent i is assigned their favorite choice among options still available
- **Lemma**. The serial dictatorship mechanism is strategyproof & Pareto optimal.
- Why is it Pareto optimal?
    - Idea: show no other assignment can Pareto dominate
    - That is, does not make anyone better off without making another worse off
    - That is, any other assignment must make some agent worse off

Serial Dictatorships (SD)

- **Lemma**. The serial dictatorship mechanism is strategyproof & Pareto optimal.
- Let M be the output of SD algorithm. Proof of Pareto-optimality:
- Let M' be any assignment where no agent is worse off than in M
    - If any agent is worse off in M' it cannot Pareto dominate M!
- **Claim**: Any such M' is identical to M, and thus M must be Pareto optimal
    - M' must give i its favorite item (which M does)
- Suppose M' is the same as M until $i = k$
- Consider agent $i = k + 1$, M gives i their favorite among remaining items
    - M' must do the same to make them not worse off
- Thus M is the unique Pareto optimal outcome

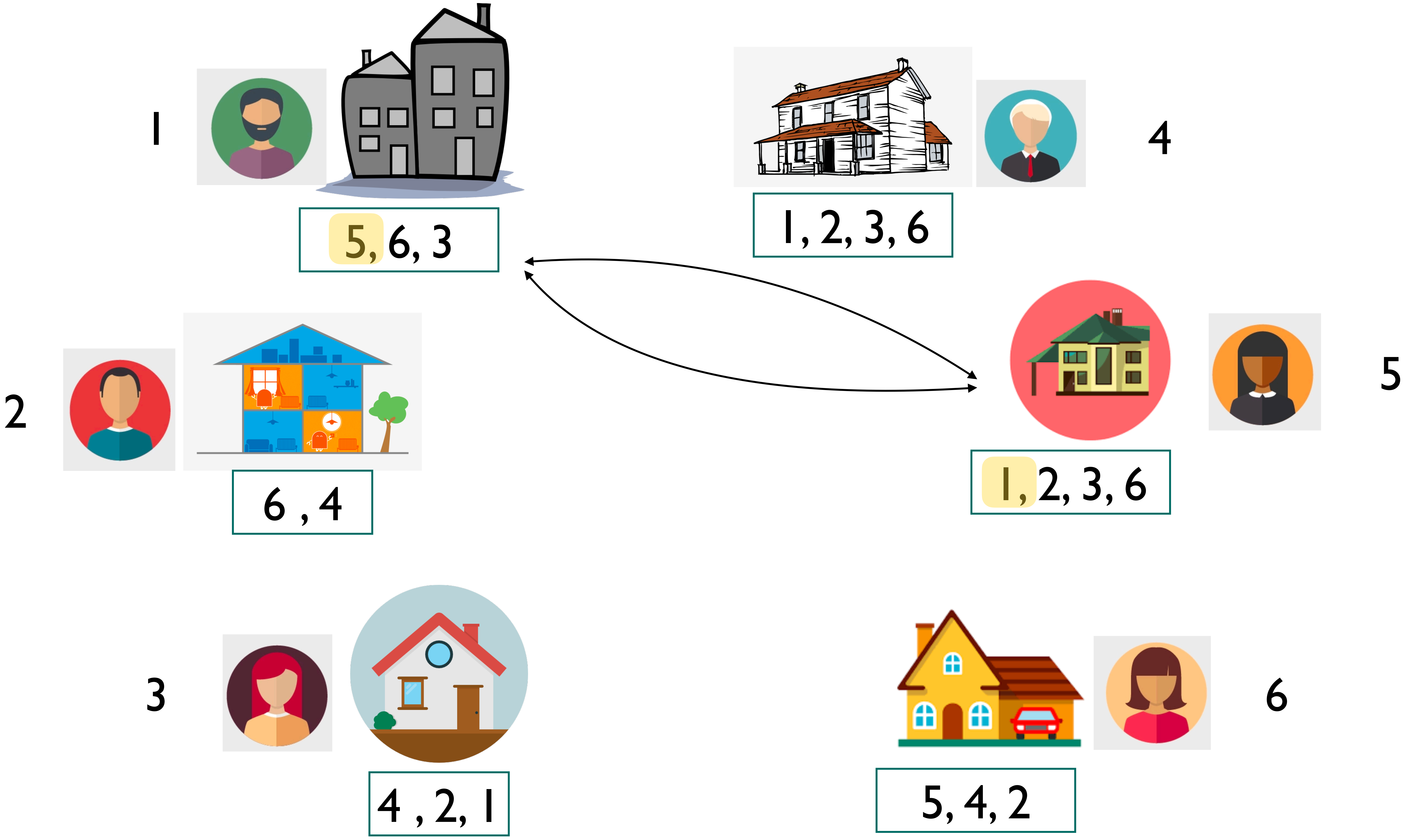
Takeways

- Serial dictatorship seems great: Pareto optimal and strategyproof
- Any criticism?
  - Can be unfair when a priority natural order between agents does not exist
- **Random-serial-dictatorship** (RSD) runs the serial-dictatorship on a ranked ordering that is sampled uniformly at random from all possible ordering
- What happens if we restrict the **# items** each agent can rank?
  - Happens in course registration (can only preregister for so many courses)
  - Truthfulness is no longer a dominant strategy:
    - Preferences now depending on the order in the lottery
- Strategizing is now all about guessing the lottery order & other's preferences

One Sided Exchange Market

- Consider n agents and n items (say houses)
- Each agent has a strict preference over the n houses
- Suppose each agent already owns one of the n houses
- Agents are willing to exchange with others to get a better one
- **Goal**. A way to reassign items to agents (perform **exchanges**) st.:
  - No one gets a house they like worse than the one they started with
  - Outcome is Pareto optimal
  - Strategyproof: truthful reporting of preferences is a dominant strategy
  - Stable / core allocation: no subset of agents can exchange amongst themselves to get a better outcome
- Sometimes called the house allocation problem

Example Instance



House Allocation Problem

- Ideas on how to design an algorithm to reallocate houses?
- Can consider all two-way swaps:
  - Are there any a, b whose favorite is the others house?
  - Can do any such swaps
- However, these many not be enough
- Sometimes we may need a three or longer trade cycle
- **Naive**: go through all 2 cycles, all 3 cycles, and so on and do any advantageous trades on those cycles
- How can we go about this systematically?

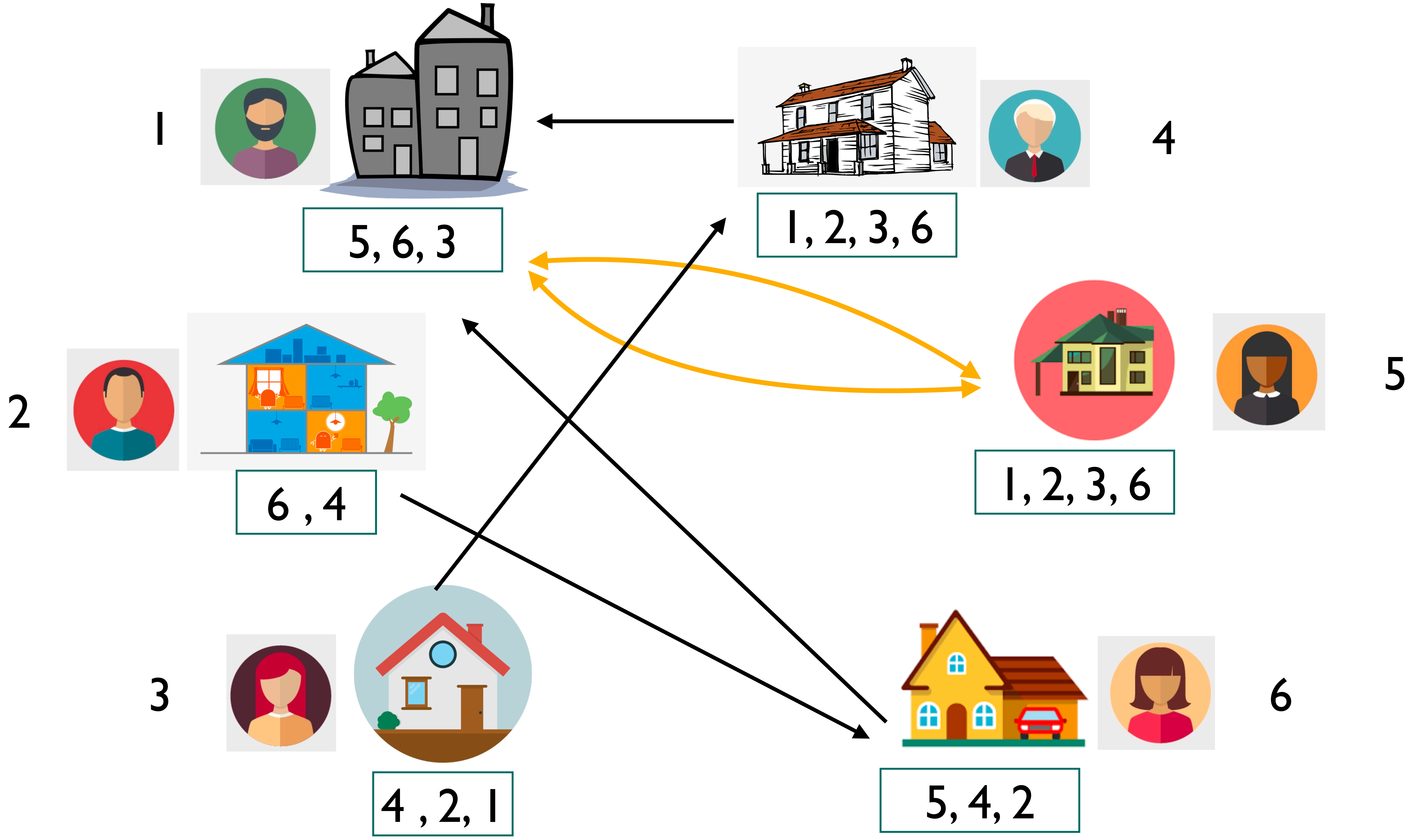
Top-Trading Cycle [Gale & Shapley]

- Each agent report their overall preferences in the beginning
- **Step 1**. Each agent (simultaneously) points to its favorite house (among houses remaining)
  - Induces a directed graph G in which every vertex has outdegree 1
  - G must have at least 1 directed cycle (self loops count)
  - Pick directed cycles and make all trades on it (each agent gives its house to the agent that points to it)
  - Delete all agents and houses that were traded in Step 1
- While agents remain, go back to **Step 1**.

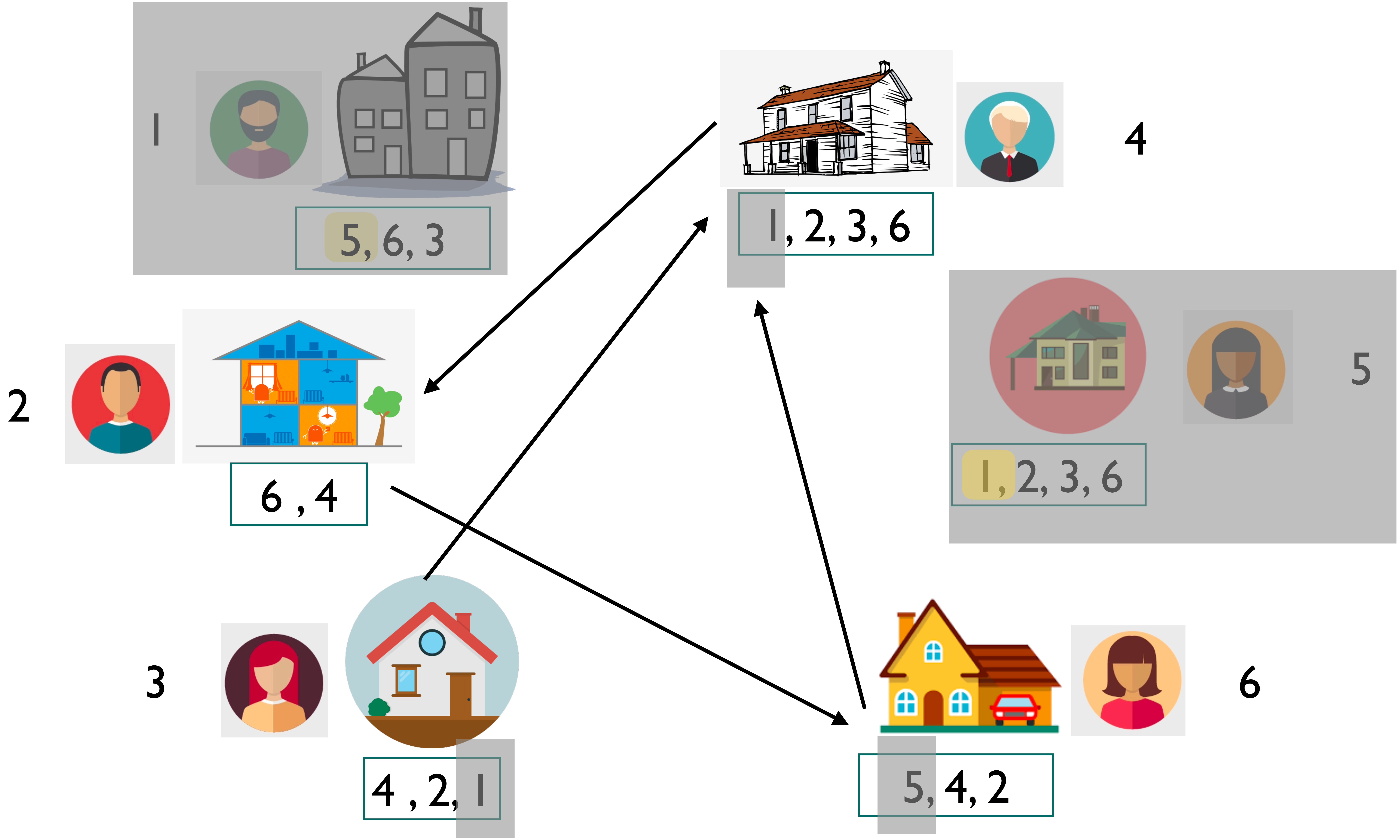
Why is there at least one directed cycle?

Can an agent be involved in two directed cycles?

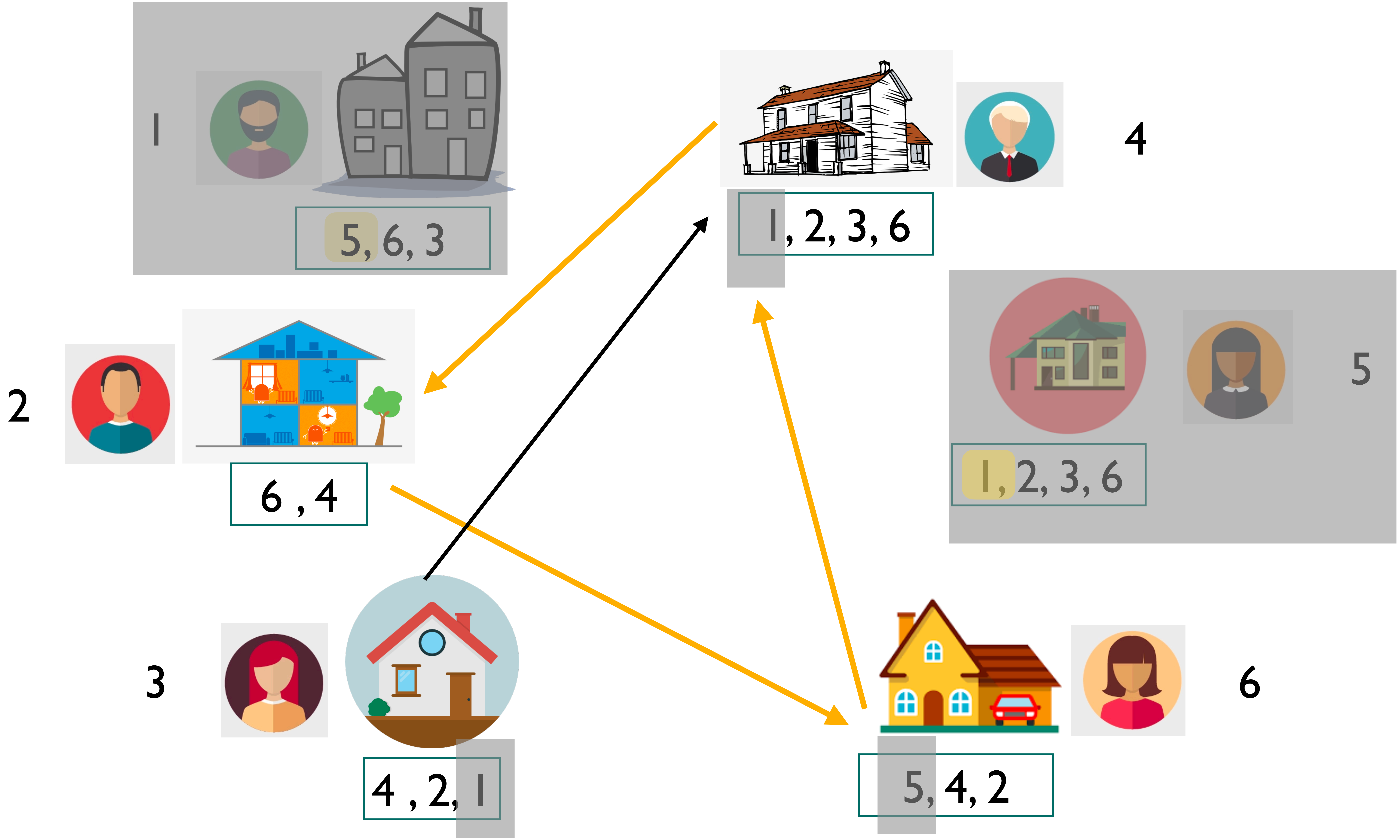
Example Instance



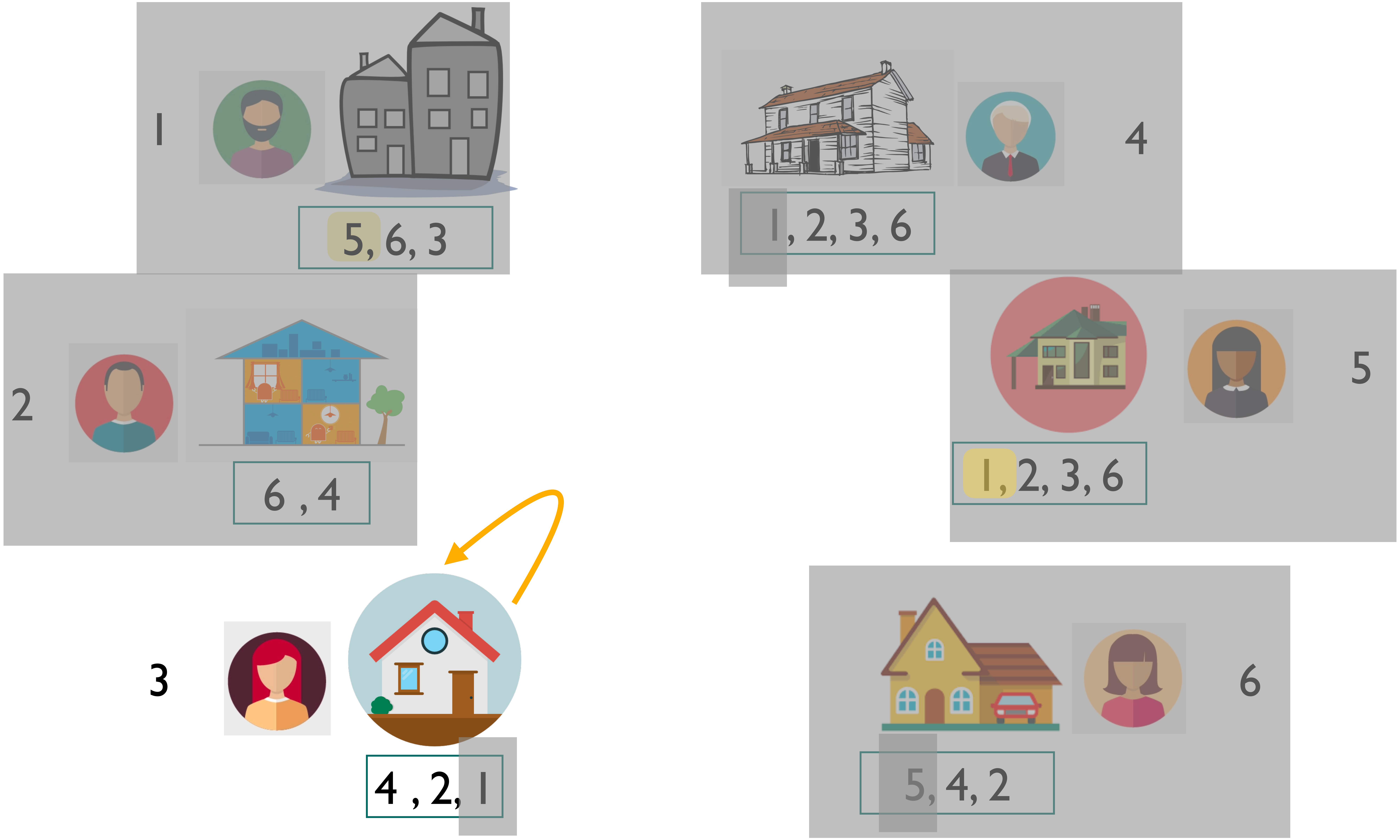
Example Instance



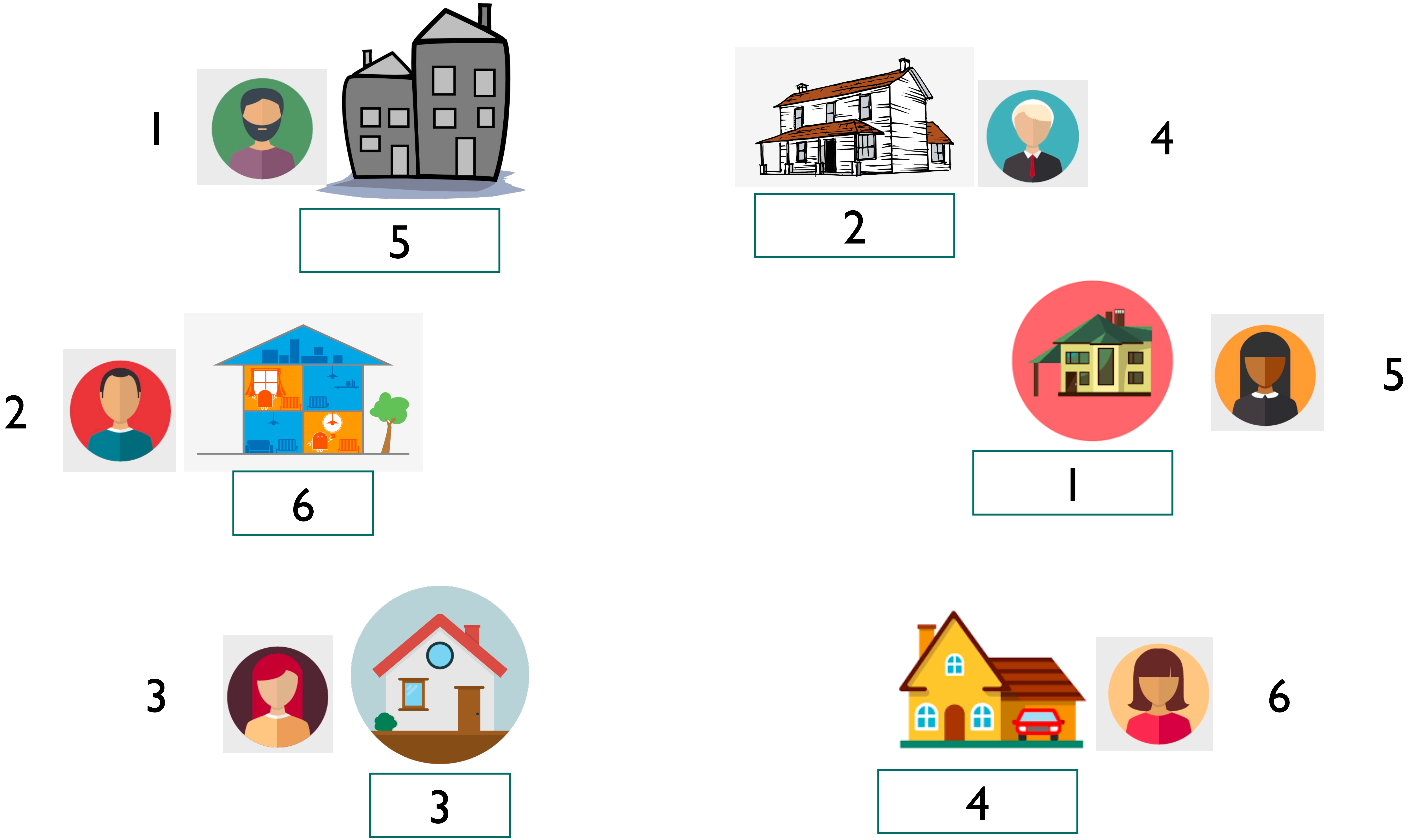
Example Instance



Example Instance



Final Output



TTC Properties

- **Time Complexity.** How many rounds until the algorithm terminates?
  - At least one trade occurs at round, at most n rounds
  - Can show that each round can be implemented in $O(n)$ time
- Everyone has an incentive to participate, that is,
  - Allocation at least as good as the one they started with, why?
  - Everyone has their own house at the end of any preference ordering
- TTC is **strategyproof (DSIC)**: being truthful is dominant strategy
  - Regardless of what other players are doing, each agent must truthfully point to their favorite remaining house in each round
  - What could be a reason to lie?
    - Point to less desirable house now to get something better in future

TTC is Strategyproof

- **Proof Overview**.
  - An agent's strategy what preference ordering over n house to submit
  - What edges are formed is pre-determined by rankings submitted
- **Goal**: Fixing everyone else's strategy s_{-i} (their rankings), show that submitting a truthful ranking gives i the best possible item
  - For any preference order i may have
  - And for any ranking of others s_{-i}
- **Claim**. At any round t, pointing truthfully at the favorite remaining house gives the best possible outcome, fixing s_{-i}

TTC is Strategyproof

- **Proof**. Consider any round t. Fix everyone else's rankings s_{-i}
- What are the choices of items that agent i can possibly get at this round?
- Let N_i be i's choice set: of set of items that have a directed path to agent i
  - That is, if i were to point to any item in N_i : a directed cycle could form
- $|N_i|$ cannot go down in round $t + 1$ if i is still unmatched
  - If agent j points to i at round t means i is their favorite among remaining items: this does not change as long as i is still unmatched
- Thus, pointing to favorite remaining item (in N_i or outside if $N_i = \emptyset$) gets best possible outcome: truthful reporting is a dominant strategy

TTC is Stable

- Given a strict preference raking by n agents let $M(i)$ denote the house they receive by running TTC
- **(Stable Allocation)**
  - A subset $S \subseteq \{1,\ldots,n\}$ is a **blocking pair** if there is a way to trade the houses $M(j)$ they receive from TTC amongst themselves to make one of them better off without making anyone else worse off
  - An allocation is **stable** is there is no such blocking pair
- Stable allocations are also called "core" allocations in the literature
- Stronger condition than Pareto optimality!
  - Implies Pareto optimality when $S = N$

Stable Allocation

- **Theorem**. TTC algorithm outputs a stable allocation.
- **Proof**. Consider an arbitrary subset S
- Let N_j denote the set of agents that get allocated in the j round in TTC
- Let ℓ be the **first round** in an agent $i \in S$ receives their house
  - i gets their favorite house among those not obtained by $N_1, \ldots, N_{\ell-1}$
  - No member of S among these, that is,
    - $N_j \cap S = \emptyset$ for $j = 1, \ldots, \ell - 1$
    - Because ℓ is the first round where anyone in S gets their house
- No reallocation within S can make i better off!

Stable Allocation

- **Theorem**. TTC algorithm outputs a **unique** stable allocation.
- **Proof**. Let N_j denote the set of agents who get allocated in round j by TTC
- All agents of N_1 receive their first choice: this must be true in any stable allocation
  - If not, the agents of N_1 can form a coalition for which internal reallocation can make everyone strictly better off
- Similarly, all agents of N_2 receive their top choice outside N_1
  - Given that every stable allocation agrees with TTC for agents in N_1, such an allocation must also agree for agents in N_2
- Inductively we can show that TTC allocation must be the unique stable allocation ∎

Summary

- TTC is a computationally efficient, strategyproof, Pareto optimal and stable allocation algorithm for exchange markets
- Given all its nice properties, we don't hear of it as much as lotteries
- How good is the algorithm for practical applications?
    - Paired-kidney donation markets
    - School assignment (even though it doesn't fit the exchange model)