CSC304
Algorithmic Game Theory & Mechanism Design

Nisarg Shah

Introduction

- **Instructor:** Nisarg Shah (~nisarg, nisarg@cs, SF 2301C)
- **TAs:** Evi Micha (emicha@cs)
  Calum MacRury (calum.macrury@gmail)
  Stephanie Knill (knill.stephanie@gmail)
- **Lectures:** Wed-Fri, 3-4pm, GB 248
- **Tutorials:** Mon, 3-4pm
  Birth month = Jan-Jun → GB 248
  Birth month = Jul-Dec → LM 155
- **Office hours:** Fri 4-5pm in SF 2301C (except today)

No tutorial next Monday (Sep 9)

First tutorial will be on Mon Sep 16. Check the course webpage for further announcements.

Course Information

- Course Page:
  `www.cs.toronto.edu/~nisarg/teaching/304f19/`

- Discussion Board:
  `piazza.com/utoronto.ca/fall2019/csc304`

- Grading – MarkUs system
  - Link will be distributed after about two weeks
  - LaTeX preferred, scans are OK!
  - An arbitrary subset of questions may be graded…

Course Organization

- Three (roughly equal) parts:
  - Game theory
  - Mechanism design with money
  - Mechanism design without money
- A homework and a midterm for each part
- Final exam = third midterm + a section on entire syllabus

Textbook

- Not really.
  - Slides will be your main reference.

- But…but…I want a textbook?
  - OK…
  - Book by Prof. David Parkes at Harvard
    - In preparation…
    - Closely follows the syllabus structure
    - Available from my webpage (username/password emailed to you)
  - Other good books mentioned in the handout

Grading Policy

- 3 homeworks * 15% = 45%
- 3 midterms * 15% = 45%
- Final exam (entire syllabus) = 10%
  - Final exam: third midterm + entire syllabus = 15+10 = 25%

Other Policies

- Collaboration
  - Individual homeworks.
  - Free to discuss with classmates or read online material.
  - Must write solutions in your own words (easier if you do not take any pictures/notes from the discussions)

- Citation
  - For each question, must cite the peer (write the name) or the online sources (provide links) referred, if any.
  - Failing to do this is plagiarism!

Other Policies

- "No Garbage" Policy
  - Borrowed from: Prof. Allan Borodin (citation!)
  1. Partial marks for viable approaches
  2. Zero marks if the answer makes no sense
  3. 20% marks if you admit to not knowing how to solve
     - 20% > 0% !!
- Applies to assignments+exams
  - To questions and even to subquestions
  - Doesn't apply to bonus questions

Other Policies

- Late Days
  - 3 late days total across 3 homeworks
  - At most 2 late days for a single homework
  - Covers legitimate reasons such as illness, University activities, etc.

Enough with the boring stuff.

What will we study?

Why will we study it?

What is this course about?

- Game Theory and Mechanism Design
  - Topics from microeconomics
- + Computer Science:
  - Algorithmic Game Theory (AGT)
  - Algorithmic Mechanism Design (AMD)

Game Theory

- How do rational, self-interested agents act?
- Each agent has a set of possible actions
- Rules of the game:
  - Rewards for the agents as a function of the actions taken by different agents

- We focus on noncooperative games
  - No external force or agencies forming coalitions

Example: Prisoner's Dilemma

John's Actions / Sam's Actions	Stay Silent	Betray
Stay Silent	(-1 , -1)	(-3 , 0)
Betray	(0 , -3)	(-2 , -2)

- What Sam thinks:
  - If John is going to stay silent…
    - Better for me to betray (my reward: 0)
    - Than for me to stay silent (my reward: -1)
  - If John is going to betray…
    - Better for me to betray (my reward: -2)
    - Than for me to stay silent (my reward: -3)

Only makes sense to betray

John thinks the same

That's cute.

But is this really useful in the real world?

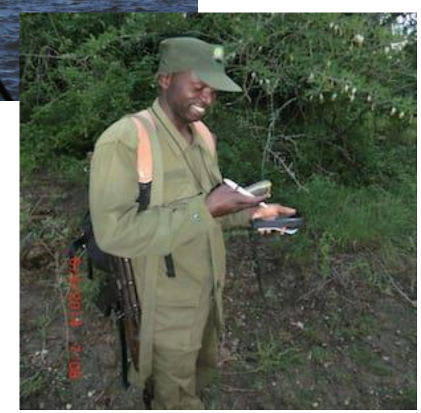
Security Games

Deploying "patrol units" to protect infrastructure targets, prevent smuggling, save wildlife…

LA Metro

LAX

Staten Island Ferry



Ugandan Forest

Image Courtesy: Teamcore

Security Games

- n targets

- Player 1: Attacker
  - Actions: attack a target

- Player 2: Defender
  - Actions: protect k ($< n$) targets at a time
  - $\binom{n}{k}$ actions – exponential!

- Attacker can observe ⇒ need to randomize
- Large games ⇒ need fast algorithms

Mechanism Design

- Design the rules of the game

- A principal in the system
  - Wants the n rational agents to behave “nicely”

- Decides the rewards (or penalties) as a function of actions to incentivize the desired behavior
  - Often the desired behavior is unclear
  - E.g., want agents to reveal their true preferences

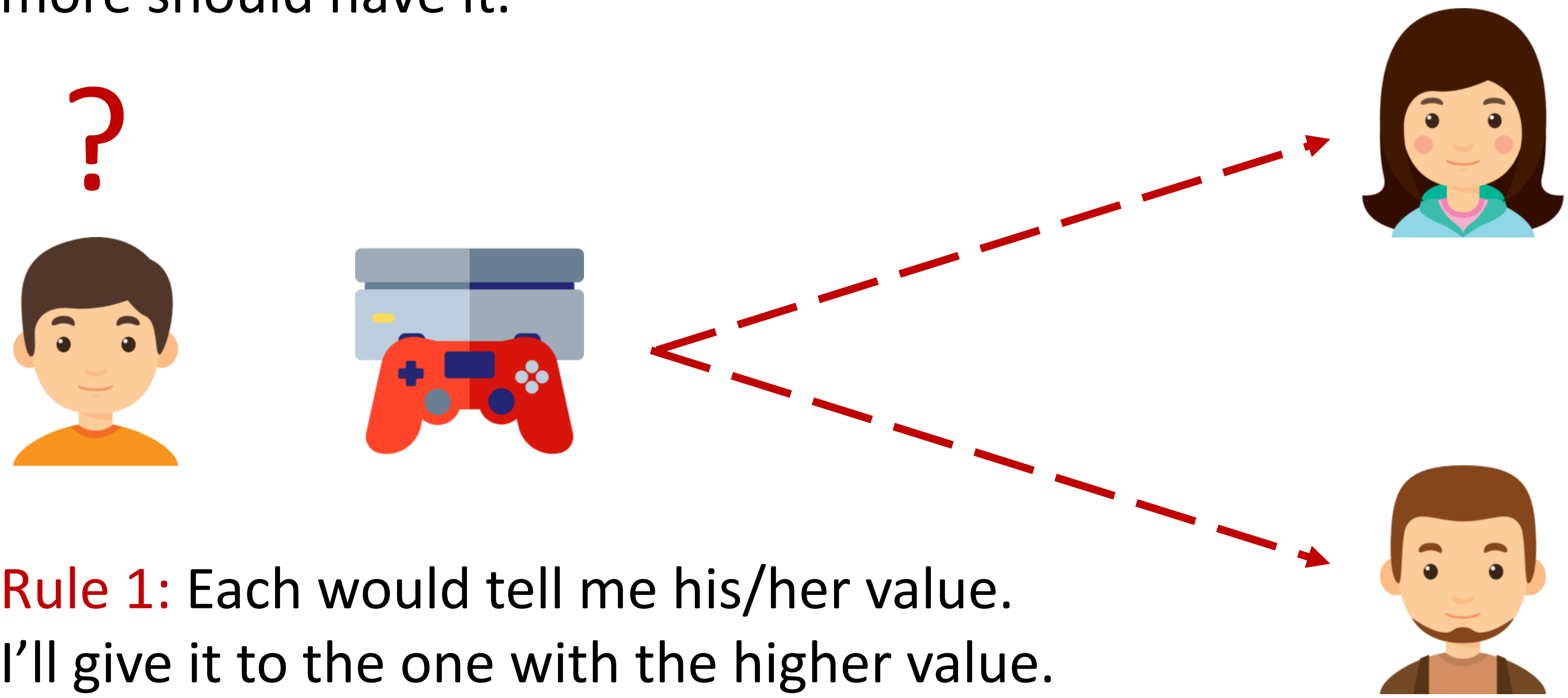
Mechanism Design

- With money
  - Principal can "charge" the agents (require payments)
  - Helps significantly
  - Example: auctions

- Without money
  - Monetary transfers are not allowed
  - Incentives must be balanced otherwise
  - Often impossible without sacrificing the objective a little
  - Example: elections, kidney exchange

Example: Auction

Objective: The one who really needs it more should have it.

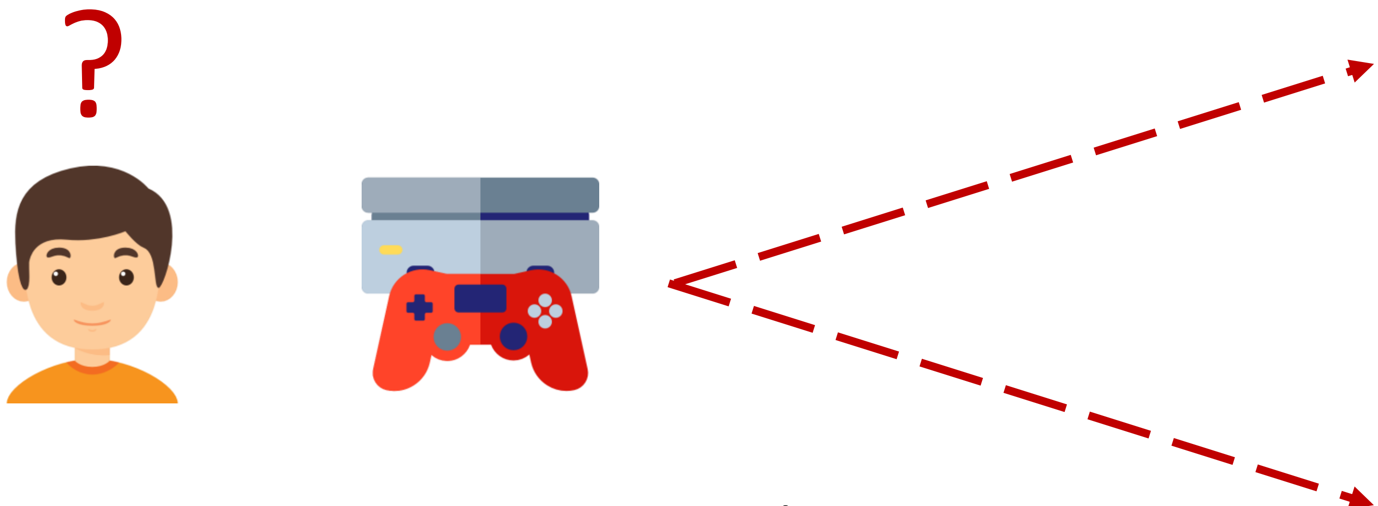


Rule 1: Each would tell me his/her value.
I'll give it to the one with the higher value.

Image Courtesy: Freepik

Example: Auction

Objective: The one who really needs it more should have it.

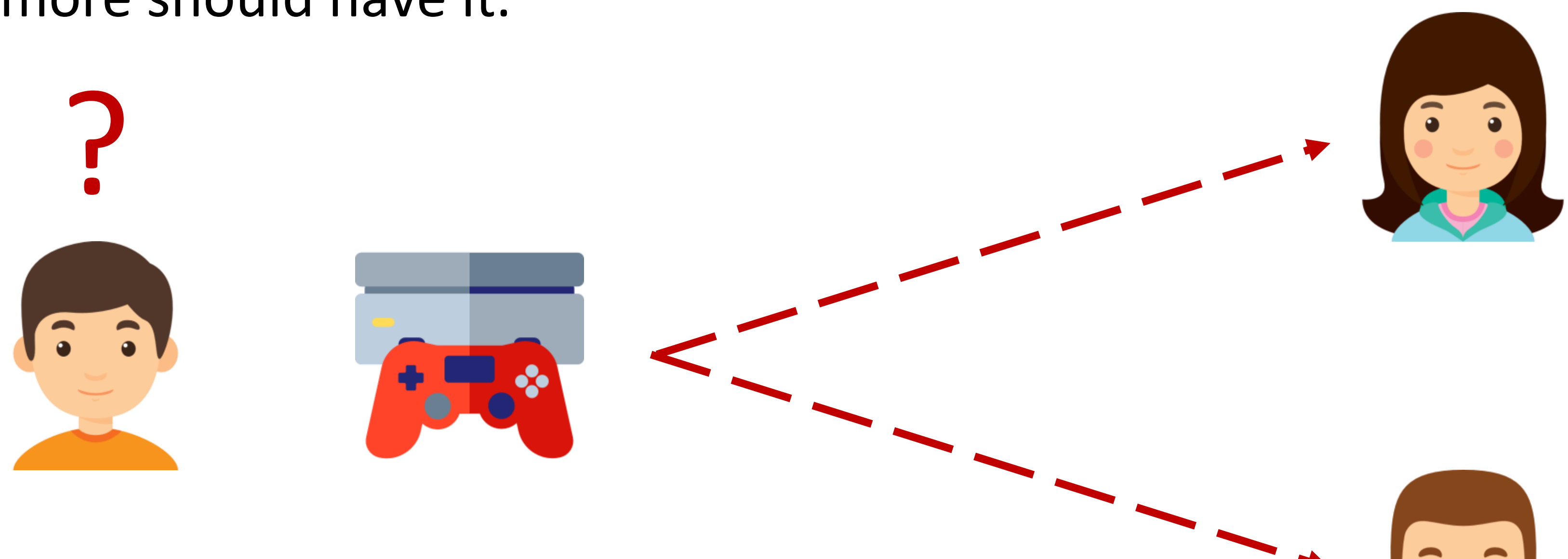


Rule 2: Each would tell me his/her value. I'll give it to the one with the higher value, but they have to pay me that value.

Image Courtesy: Freepik

Example: Auction

Objective: The one who really needs it more should have it.



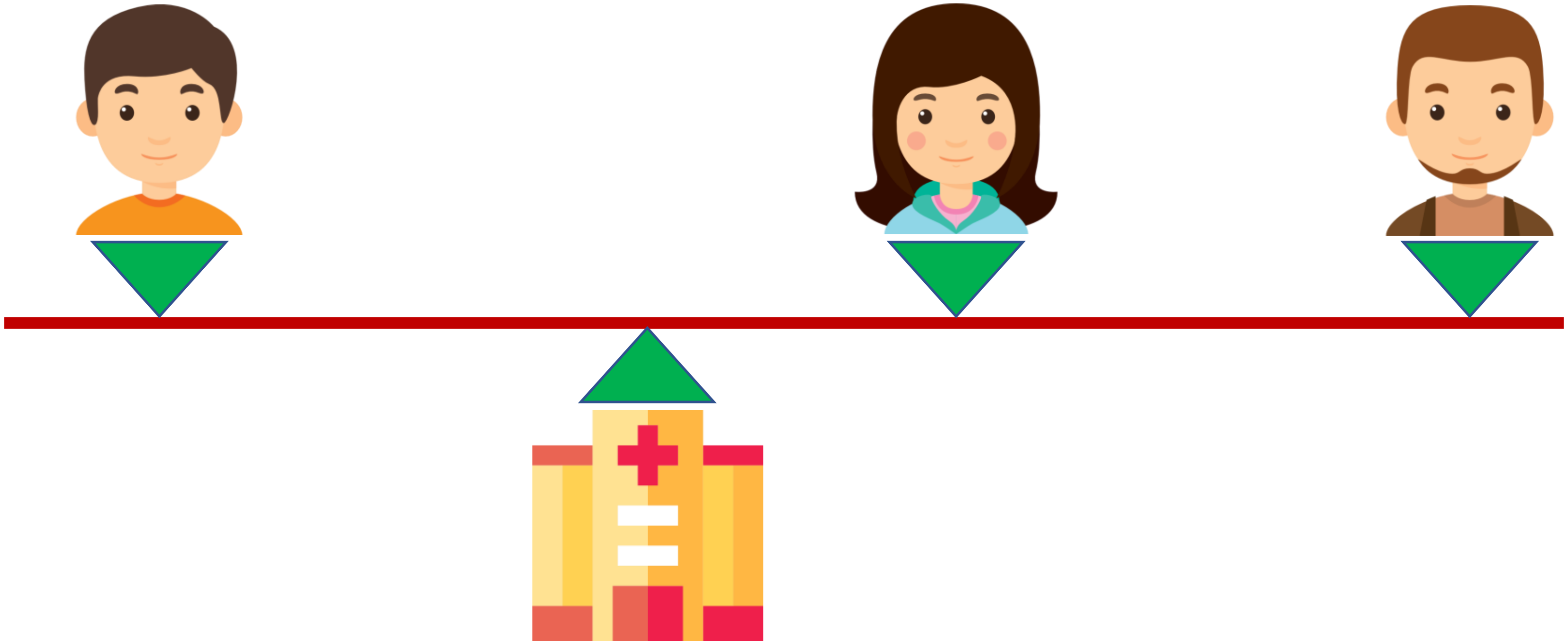
Can I make it easier so that each can just truthfully tell me how much they value it?

Image Courtesy: Freepik

Real-World Applications

- Auctions form a significant part of mechanism design with money
- Auctions are ubiquitous in the real world!
  - A significant source of revenue for many large organizations (including Facebook and Google)
  - Often run billions of tiny auctions everyday
  - Need the algorithms to be fast

Example: Facility Location



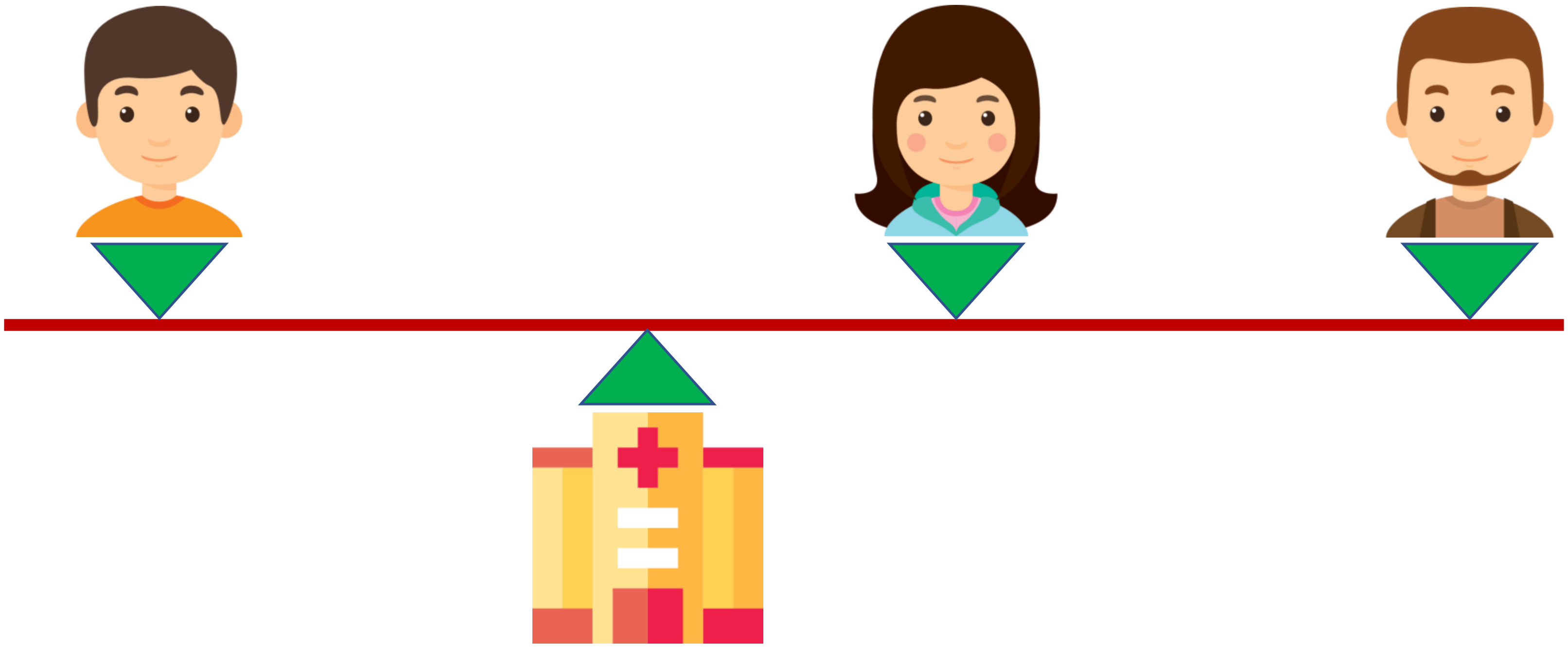
Cost to each agent: Distance from the hospital

Objective: Minimize the sum of costs

Constraint: No money

Image Courtesy: Freepik

Example: Facility Location

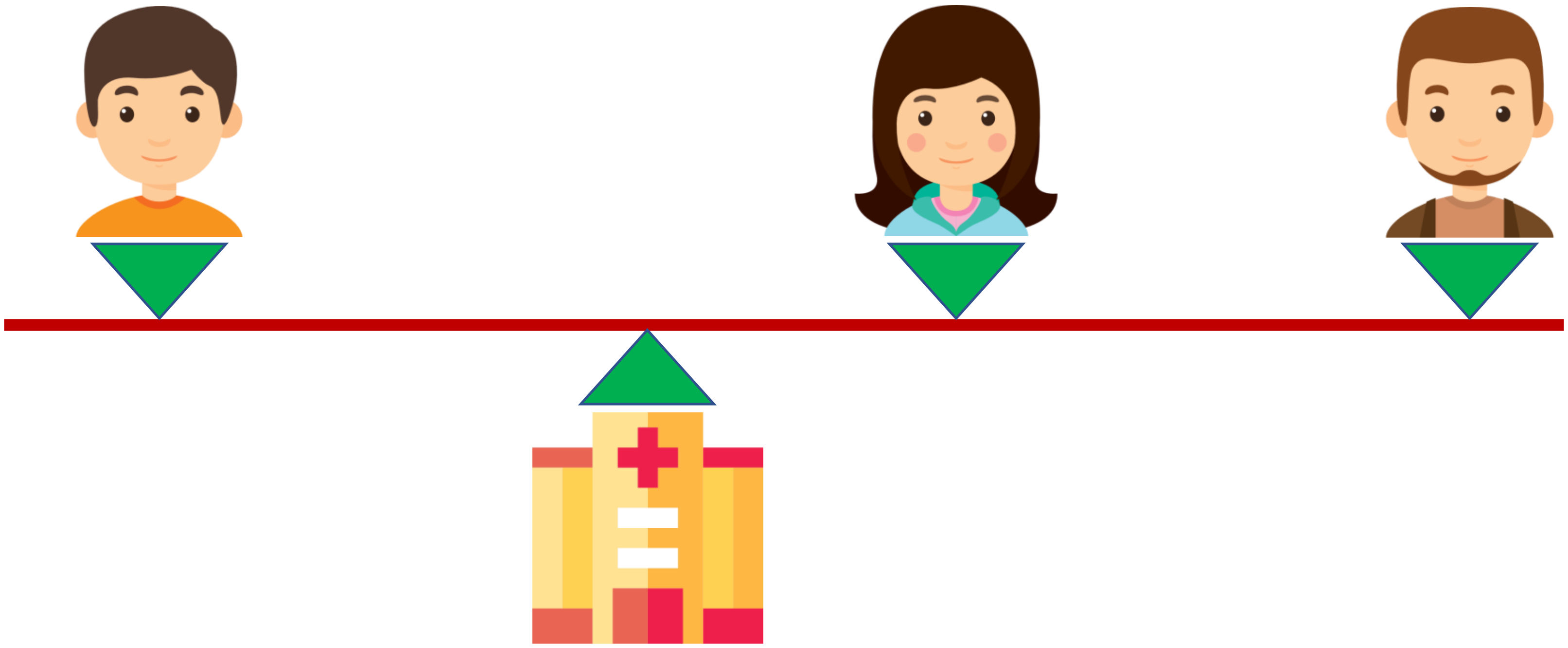


Q: What is the optimal hospital location?

Q: If we decide to choose the optimal location, will the agents really tell us where they live?

Image Courtesy: Freepik

Example: Facility Location



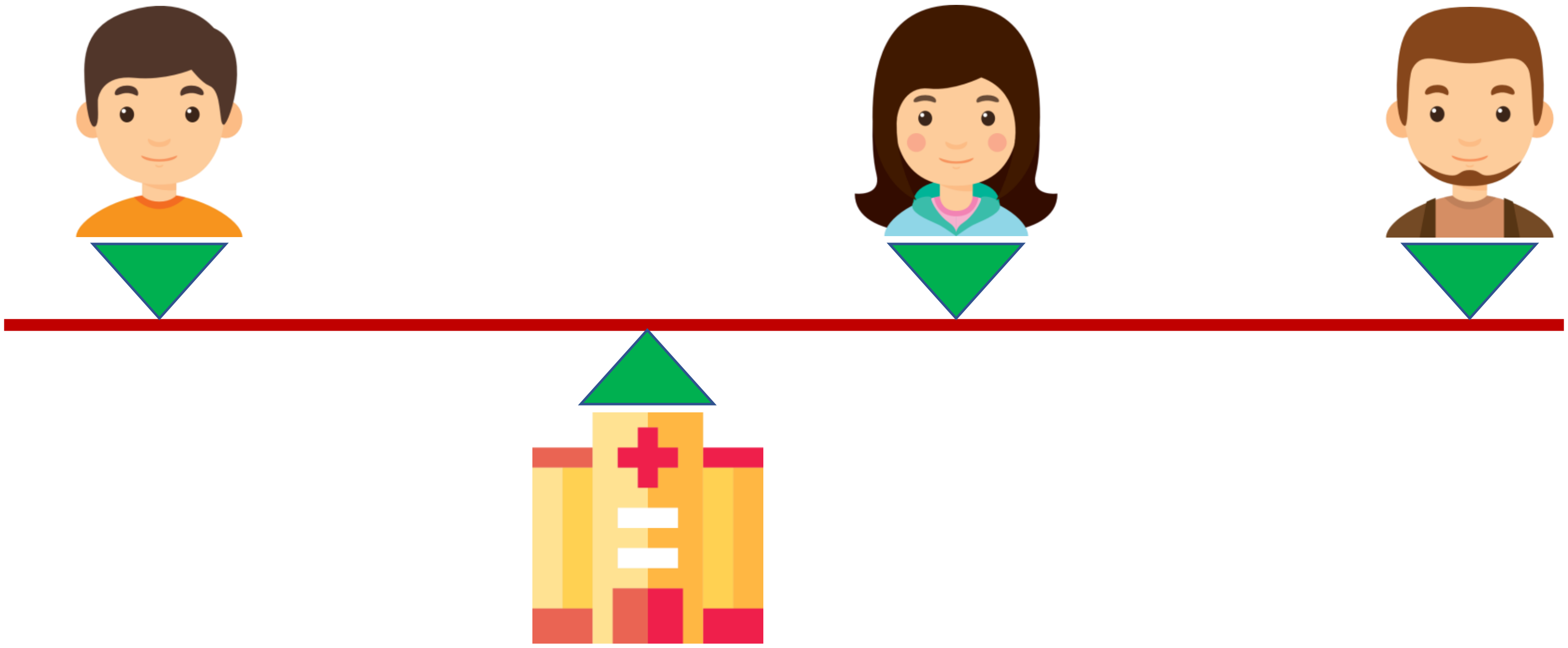
Cost to each agent: Distance from the hospital

Objective: Minimize the maximum cost

Constraint: No money

Image Courtesy: Freepik

Example: Facility Location



Q: What is the optimal hospital location?

Q: If we decide to choose the optimal location, will the agents really tell us where they live?

Image Courtesy: Freepik

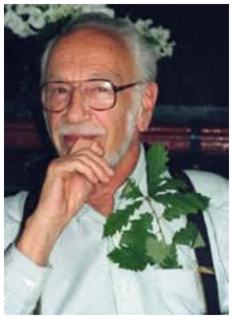
Mechanism Design w/o Money

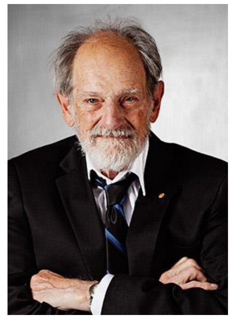
- Truth-telling is not the only possible desideratum
  - Fairness
  - Stability
  - Efficiency
  - …

- Consequently, many subfields of study
  - Fair allocation of resources
  - Stable matching
  - Voting

Real-World Applications


Roth


Gale


Shapley

National Resident Matching Program (NRMP)

School Choice (New York, Boston)

Fair Division

spliddit

Voting

ROBOVOTE