CSE 421 Section 1

Stable Matchings and Proofs Workshop

Administrivia and introductions

Your Section TA

- Runs your section
- Default TA for general questions

Your photo here

OH: [time/day/location]

Email: [user]@cs.washington.edu

All Course TAs

- Homework/exam grading
- Office hours and Ed questions

Ajay Harilal

Ben Zhang

Daniel Gao

Edward Qin

George King

Glenn Sun

Owen Boseley

Paul Han

Announcements

- **Section materials**
    - Handouts will be provided in each section
    - Solutions and slides on course webpage the evening after section

- **HW1**
    - Due Wednesday, 10/2 @ 11:59pm

Homework

LaTeX (preferred)	**Google Docs/Word**	**Handwritten**
• overleaf.com • Template available • Ask us for syntax help	• Use equation editor for math and variables	• Write neatly • Great for diagrams • Use B/W scanning app

No matter what format...

- Turn in via Gradescope
- Due Wednesdays at 11:59pm (except around Thanksgiving)

Late problems policy (NOT assignments)

- Up to **10 total problem late days**
- Use up to **2 late days per problem**
- Each part of a late day counts as a day

Stable matchings

Stable matching problem

Input: Two sets P and R of n people each, with each person having a preference list for members of the other group

Output: A stable matching between the two groups

Stable matching: perfect matching with no unstable pairs

everyone matched to exactly one person from other group

two people who prefer each other to their current matches

Gale–Shapley algorithm

We call P the **proposers** and R the **receivers**.

1. Initialize the status of all $p \in P$ and $r \in R$ to free.
2. While there is a free $p \in P$,
    a. Let r be the highest person on p's list that p has not yet proposed to.
    b. If r is free,
        i. Match p and r.
    c. Otherwise, if r prefers p over their current match p',
        i. Unmatch p' and r.
        ii. Match p and r.

Problem 1 – Gale–Shapley review

$p_1 : r_3 > r_1 > r_2 > r_4$
$p_2 : r_2 > r_1 > r_4 > r_3$
$p_3 : r_2 > r_3 > r_1 > r_4$
$p_4 : r_3 > r_4 > r_1 > r_2$

$r_1 : p_4 > p_1 > p_3 > p_2$
$r_2 : p_1 > p_3 > p_2 > p_4$
$r_3 : p_1 > p_3 > p_4 > p_2$
$r_4 : p_3 > p_1 > p_2 > p_4$

1. Initialize the status of all $p \in P$ and $r \in R$ to free.
2. While there is a free $p \in P$,
   a. Let r be the highest person on p's list that p has not yet proposed to.
   b. If r is free,
      i. Match p and r.
   c. Otherwise, if r prefers p over their current match p',
      i. Unmatch p' and r.
      ii. Match p and r.

a) Run the Gale–Shapley algorithm on the instance shown. When multiple p_i are free to propose, choose the one with the **smallest** index (e.g., if p_1 and p_2 are both free, have p_1 propose).

Taking 8 volunteers!

Solution

Problem 1 – Gale–Shapley review

a) Run the Gale–Shapley algorithm on the instance shown. When multiple p_i are free to propose, choose the one with the **smallest** index (e.g., if p_1 and p_2 are both free, have p_1 propose).

$p_1 : r_3 > r_1 > r_2 > r_4$
$p_2 : r_2 > r_1 > r_4 > r_3$
$p_3 : r_2 > r_3 > r_1 > r_4$
$p_4 : r_3 > r_4 > r_1 > r_2$

$r_1 : p_4 > p_1 > p_3 > p_2$
$r_2 : p_1 > p_3 > p_2 > p_4$
$r_3 : p_1 > p_3 > p_4 > p_2$
$r_4 : p_3 > p_1 > p_2 > p_4$

p_1 chooses r_3	(p_1, r_3)
p_2 chooses r_2	$(p_1, r_3), (p_2, r_2)$
p_3 chooses r_2	$(p_1, r_3), \cancel{(p_2, r_2)}, (p_3, r_2)$
p_2 chooses r_1	$(p_1, r_3), (p_2, r_1), (p_3, r_2)$
p_4 chooses r_3	$(p_1, r_3), (p_2, r_1), (p_3, r_2)$, (p_4, r_3) fails
p_4 chooses r_4	$(p_1, r_3), (p_2, r_1), (p_3, r_2), (p_4, r_4)$

Problem 1 – Gale–Shapley review

$p_1 : r_3 > r_1 > r_2 > r_4$

$p_2 : r_2 > r_1 > r_4 > r_3$

$p_3 : r_2 > r_3 > r_1 > r_4$

$p_4 : r_3 > r_4 > r_1 > r_2$

$r_1 : p_4 > p_1 > p_3 > p_2$

$r_2 : p_1 > p_3 > p_2 > p_4$

$r_3 : p_1 > p_3 > p_4 > p_2$

$r_4 : p_3 > p_1 > p_2 > p_4$

1. Initialize the status of all $p \in P$ and $r \in R$ to free.
2. While there is a free $p \in P$,
   a. Let r be the highest person on p's list that p has not yet proposed to.
   b. If r is free,
      i. Match p and r.
   c. Otherwise, if r prefers p over their current match p',
      i. Unmatch p' and r.
      ii. Match p and r.

b) What if you default to the one with the **largest** index? Does the answer change?

c) What if the r_i propose instead of the p_i? Does the answer change?

Try it yourself, or with people near you!

Solution

Problem 1 – Gale–Shapley review

$p_1 : r_3 > r_1 > r_2 > r_4$
$p_2 : r_2 > r_1 > r_4 > r_3$
$p_3 : r_2 > r_3 > r_1 > r_4$
$p_4 : r_3 > r_4 > r_1 > r_2$

$r_1 : p_4 > p_1 > p_3 > p_2$
$r_2 : p_1 > p_3 > p_2 > p_4$
$r_3 : p_1 > p_3 > p_4 > p_2$
$r_4 : p_3 > p_1 > p_2 > p_4$

1. Initialize the status of all $p \in P$ and $r \in R$ to free.
2. While there is a free $p \in P$,
    a. Let r be the highest person on p's list that p has not yet proposed to.
    b. If r is free,
        i. Match p and r.
    c. Otherwise, if r prefers p over their current match p',
        i. Unmatch p' and r.
        ii. Match p and r.

b) What if you default to the one with the **largest** index? Does the answer change?

No change!

c) What if the r_i propose instead of the p_i? Does the answer change?

Different result!

Turns out, (b) is always true! You will prove it in lecture tomorrow.

Solution

Problem 1 – Gale–Shapley review

b) What if you default to the one with the **largest** index? Does the answer change?

$$
\begin{aligned}
p_1 &: r_3 > r_1 > r_2 > r_4 \\
p_2 &: r_2 > r_1 > r_4 > r_3 \\
p_3 &: r_2 > r_3 > r_1 > r_4 \\
p_4 &: r_3 > r_4 > r_1 > r_2 \\
\\
r_1 &: p_4 > p_1 > p_3 > p_2 \\
r_2 &: p_1 > p_3 > p_2 > p_4 \\
r_3 &: p_1 > p_3 > p_4 > p_2 \\
r_4 &: p_3 > p_1 > p_2 > p_4
\end{aligned}
$$

p_4 chooses r_3	(p_4, r_3)
p_3 chooses r_2	$(p_3, r_2), (p_4, r_3)$
p_2 chooses r_2	$(p_3, r_2), (p_4, r_3)$, (p_2, r_2) fails
p_2 chooses r_1	$(p_2, r_1), (p_3, r_2), (p_4, r_3)$
p_1 chooses r_3	$(p_1, r_3), (p_2, r_1), (p_3, r_2), \cancel{(p_4, r_3)}$
p_4 chooses r_4	$(p_1, r_3), (p_2, r_1), (p_3, r_2), (p_4, r_4)$

Solution

Problem 1 – Gale–Shapley review

c) What if the r_i propose instead of the p_i? Does the answer change?

$p_1 : r_3 > r_1 > r_2 > r_4$
$p_2 : r_2 > r_1 > r_4 > r_3$
$p_3 : r_2 > r_3 > r_1 > r_4$
$p_4 : r_3 > r_4 > r_1 > r_2$

$r_1 : p_4 > p_1 > p_3 > p_2$
$r_2 : p_1 > p_3 > p_2 > p_4$
$r_3 : p_1 > p_3 > p_4 > p_2$
$r_4 : p_3 > p_1 > p_2 > p_4$

r_1 chooses p_4	(p_4, r_1)
r_2 chooses p_1	$(p_1, r_2), (p_4, r_1)$
r_3 chooses p_1	$(p_1, r_3), \cancel{(p_1, r_2)}, (p_4, r_1)$
r_2 chooses p_3	$(p_1, r_3), (p_3, r_2), (p_4, r_1)$
r_4 chooses p_3	$(p_1, r_3), (p_3, r_2), (p_4, r_1)$, (p_3, r_4) fails
r_4 chooses p_1	$(p_1, r_3), (p_3, r_2), (p_4, r_1)$, (p_1, r_4) fails
r_4 chooses p_2	$(p_1, r_3), (p_2, r_4), (p_3, r_2), (p_4, r_1)$

Problem 2 – Number of stable matchings

We saw an instance of stable matching with two stable matchings.

Is there an instance with more than two? Give example (if yes) or proof (if no).

Take 3 minutes to brainstorm with the people around you, then we'll discuss.

Solution

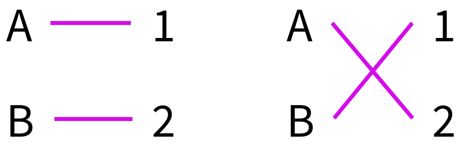
Problem 2 – Number of stable matchings

Is there an instance with more than two? Give example (if yes) or proof (if no).

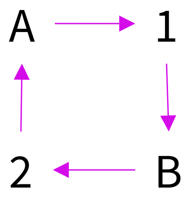
Try smaller examples. What's an easy instance with two stable matchings?

$$\boxed{\begin{array}{l} A : 1 > 2 \\ B : 2 > 1 \\ \\ 1 : B > A \\ 2 : A > B \end{array}}$$

both matchings stable



"square preference cycle"



$\longrightarrow$ = prefers

Solution

Problem 2 – Number of stable matchings

Is there an instance with more than two? Give example (if yes) or proof (if no).

Now generalize to three. One possible solution:

$A : 1 > 2 > 3$

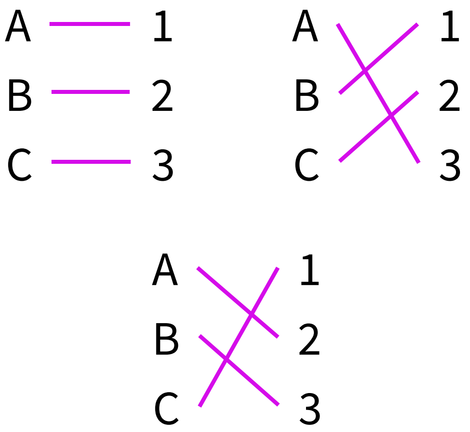
$B : 2 > 3 > 1$

$C : 3 > 1 > 2$

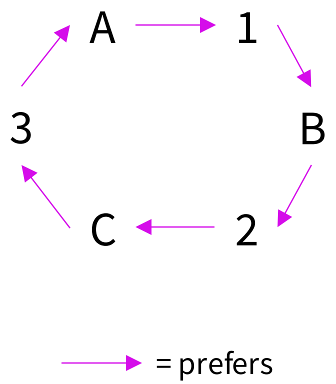
$1 : B > C > A$

$2 : C > A > B$

$3 : A > B > C$



"hexagon preference cycle"



Proof-writing workshop

Graph theory review

- **degree:** number of edges connected to a vertex
- **path** (walk): list of vertices $v_1, v_2, \ldots, v_k$ such that each $\{v_i, v_{i+1}\}$ is an edge
  - for directed graphs, (v_i, v_{i+1})
- **cycle** (closed walk): path with same first and last vertex
- **simple path** (path): path with all distinct vertices
- **simple cycle** (cycle): cycle with all distinct vertices, except first/last
- **connected:** there is a path between any two vertices
- **tree:** connected acyclic (no cycles) graph
- **rooted tree:** tree with a designated vertex called the root
  - Note that “parent” and “child” are not defined unless the tree is rooted!

Problem 3 – Proof-writing workshop

In this problem, you will **read many proofs** of the following claim:

Claim. Every tree with at least 2 vertices has at least 2 vertices of degree 1.

a) First, take 3 minutes to think about the problem yourself. How would you prove it?

Problem 3 – Proof-writing workshop

Qualities of a good proof

Correct	Complete	Concise	Clear
• No false statements	• Claims justified • Hypotheses used • Notation defined	• No excessive details • No unnecessary notation	• Main ideas are evident • Good stylistic choices

b) Read Sample Solution 1. Discuss with people around you — is it clear, complete, concise, clear? What would you change?

Problem 3 – Proof-writing workshop

Qualities of a good proof

Correct	Complete	Concise	Clear
• No false statements	• Claims justified • Hypotheses used • Notation defined	• No excessive details • No unnecessary notation	• Main ideas are evident • Good stylistic choices

b) Read Sample Solution 2. Discuss with people around you — is it clear, complete, concise, clear? What would you change?

Problem 3 – Proof-writing workshop

Qualities of a good proof

Correct	Complete	Concise	Clear
• No false statements	• Claims justified • Hypotheses used • Notation defined	• No excessive details • No unnecessary notation	• Main ideas are evident • Good stylistic choices

b) Read Sample Solution 3. Discuss with people around you — is it clear, complete, concise, clear? What would you change?

Problem 3 – Proof-writing workshop

Qualities of a good proof

Correct	Complete	Concise	Clear
• No false statements	• Claims justified • Hypotheses used • Notation defined	• No excessive details • No unnecessary notation	• Main ideas are evident • Good stylistic choices

b) Read Sample Solution 4. Discuss with people around you — is it clear, complete, concise, clear? What would you change?

Summary

- When stuck, look for **small examples**.
- When writing a proof, revise it to be **correct, complete, concise, and clear**.

Thanks for coming to section this week!