Stable Roommates

Winter School On COSCOE @ IIT Jodhpur

December 8, 2023

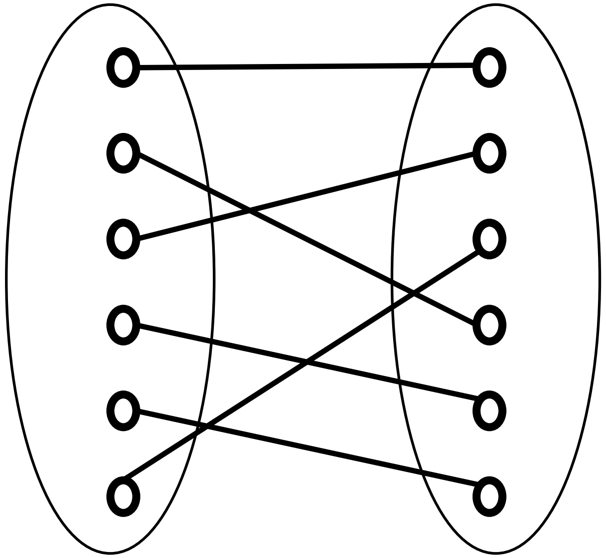
Lawqueen Kanesh

IIT Jodhpur

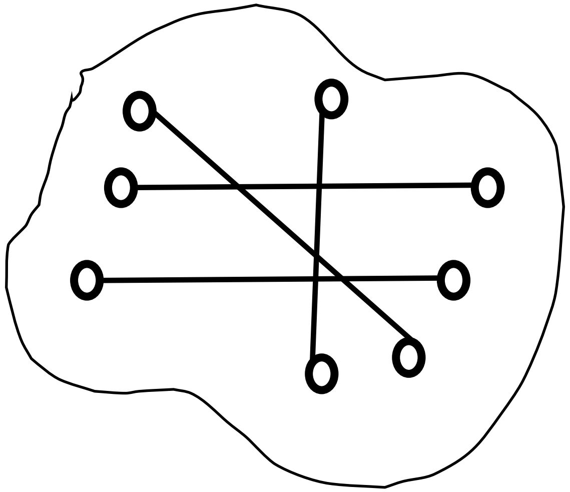
lawqueen@iitj.ac.in

Stable Roommates

- A variant of Stable Marriage problem
- Instead of having two disjoint sets (of same sizes), Stable Roommates matches people from within the same set (of even size)



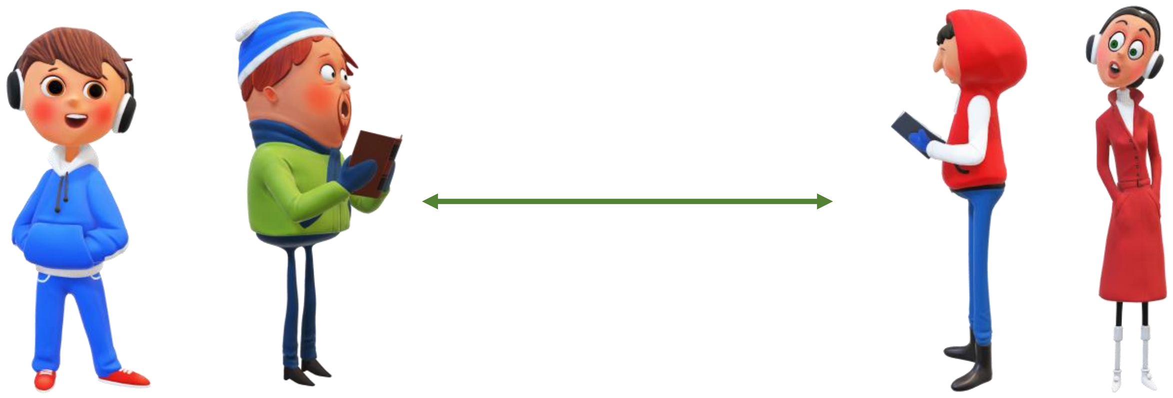
Stable Marriage



Stable Roommates

Stable Roommates

- Given a set of participants. Each of the participant ranks the others in a strict order of preference.
- Goal is to find a stable matching of the given set of people
- Stable: No blocking pair (prefer each other over matched partners)



Stable Roommates

- Stable: No blocking pair (prefer each other over matched partners)

A	**B**	**D**	**C**
B	**A**	**C**	**D**
C	**A**	**D**	**B**
D	**C**	**B**	**A**

(A,B), (C,D) is a stable matching
Unmatched pairs (A,C), (A,D), (B,C), (B,D) are not blocking pairs

Stable Roommates

- Unlike Stable Marriage, there need not exist a stable matching in the Stable Roommates problem

A	**B**	**C**	**D**
B	**C**	**A**	**D**
C	**A**	**B**	**D**
D	**(Arbitrary)**		

Whoever is paired with D prefers someone who also prefers them.

Stable Roommates

- An Efficient Algorithm for the “Stable Roommates” Problem- Robert W. Irving, 1985 $\mathcal{O}(n^2)$-running time

- Runs in two Phases

Phase 1

First phase consisting of a sequence of “proposals”

Step 1: Proposal: 1. If x receives a proposal from y, then:

(a) x rejects it at once if x already holds a better proposal;

Phase 1

First phase consisting of a sequence of "proposals"

Step 1: Proposal: 1. If x receives a proposal from y, then:

(a) x rejects it at once if x already holds a better proposal;

(b) x holds it for consideration otherwise, x rejects any poorer proposal x currently hold.

Phase 1

First phase consisting of a sequence of “proposals”

Step 1: Proposal: 1. If x receives a proposal from y, then:

(a) x rejects it at once if x already holds a better proposal;

(b) x holds it for consideration otherwise, x rejects any poorer proposal x currently hold.

2. An individual x proposes to the others in the order in which they appear in x’s preference list, stopping when a promise of consideration is received; any subsequent rejection causes x to continue immediately with the sequence of proposals

Phase 1

First phase consisting of a sequence of “proposals”

Step 1: Proposal: 1. If x receives a proposal from y, then:

(a) x rejects it at once if x already holds a better proposal;

(b) x holds it for consideration otherwise, x rejects any poorer proposal x currently hold.

2. An individual x proposes to the others in the order in which they appear in x’s preference list, stopping when a promise of consideration is received; any subsequent rejection causes x to continue immediately with the sequence of proposals

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A	**B**	**D**	**F**	**E**	**C**
B	**E**	**C**	**F**	**D**	**A**
C	**B**	**F**	**E**	**A**	**D**
D	**C**	**A**	**F**	**B**	**E**
E	**B**	**A**	**F**	**C**	**D**
F	**A**	**D**	**E**	**C**	**B**

Name	Proposed To	Proposed By
A		
B		
C		
D		
E		
F		

Phase 1

A	**B**	**D**	**F**	**E**	**C**
B	**E**	**C**	**F**	**D**	**A**
C	**B**	**F**	**E**	**A**	**D**
D	**C**	**A**	**F**	**B**	**E**
E	**B**	**A**	**F**	**C**	**D**
F	**A**	**D**	**E**	**C**	**B**

Name	Proposed To	Proposed By
A	B	
B		A
C		
D		
E		
F		

A proposes to B and B holds it for consideration

Phase 1

A	B	D	F	E	C
B	E	C	F	D	A
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	B	
B	E	A
C		
D		
E		B
F		

B proposes to E and E holds it for consideration

Phase 1

A	B	D	F	E	C
B	E	C	F	D	A
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	B	
B	E	A,C
C	B	
D		
E		B
F		

C proposes to B
B is holding A's proposal
B prefers C over A

Phase 1

A	**B**	**D**	**F**	**E**	**C**
B	**E**	**C**	**F**	**D**	**A**
C	**B**	**F**	**E**	**A**	**D**
D	**C**	**A**	**F**	**B**	**E**
E	**B**	**A**	**F**	**C**	**D**
F	**A**	**D**	**E**	**C**	**B**

Name	Proposed To	Proposed By
A	B	
B	E	A,C
C	B	
D		
E		B
F		

C proposes to B
B rejects A's proposal
Delete B from A's list and delete A from B's list

Phase 1

A		D	F	E	C
B	E	C	F	D	
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A		
B	E	C
C	B	
D		
E		B
F		

C proposes to B
B rejects A's proposal
Delete B from A's list and delete A from B's list
B holds C's proposal for consideration

Phase 1

A		D	F	E	C
B	E	C	F	D	
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	C
C	B	
D		A
E		B
F		

A proposes to D
D holds A's proposal for consideration

Phase 1

A		D	F	E	C
B	E	C	F	D	
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	C
C	B	D
D	C	A
E		B
F		

D proposes to C

C holds D's proposal for consideration

Phase 1

A		D	F	E	C
B	E	C	F	D	
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	C,E
C	B	D
D	C	A
E	B	B
F		

E proposes to B
B is holding C’s proposal
B prefers E over C

Phase 1

A		D	F	E	C
B	E	C	F	D	
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	C,E
C	B	D
D	C	A
E	B	B
F		

E proposes to B
B rejects C's proposal
Delete B from C's list and delete C from B's list
B holds E's proposal for consideration

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	E
C		D
D	C	A
E	B	B
F		

E proposes to B
B rejects C's proposal
Delete B from C's list and delete C from B's list
B holds E's proposal for consideration

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	
B	E	E
C	F	D
D	C	A
E	B	B
F		

C proposes to F
F holds C's proposal for consideration

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

F proposes to A
A holds F's proposal for consideration

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

F proposes to A
A holds F's proposal for consideration

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F	E	C
B	E		F	D	
C		F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E		F	D	
C		F	E		D
D	C	A	F	B	E
E	B		F	C	D
F	A	D	E	C	B

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F			
B	E		F	D		
C		F	E		D	
D	C	A	F	B	E	
E	B		F	C	D	
F	A	D	E	C	B	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E				
C		F	E		D
D	C	A	F		E
E	B		F	C	D
F	A	D	E	C	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E				
C		F	E		D
D	C	A	F		E
E	B		F	C	D
F	A	D	E	C	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E				
C		F	E		D
D	C	A			
E	B		F	C	
F	A		E	C	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E				
C		F	E		D
D	C	A			
E	B		F	C	
F	A		E	C	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 1

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

Name	Proposed To	Proposed By
A	D	F
B	E	E
C	F	D
D	C	A
E	B	B
F	A	C

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Phase 2

- Eliminate Rotations and reduce table size
- Iterative process that will result in either one person running out of people to propose to (in which case no stable matching exists), or every preference list shrinking to a single person, in which case they specify a stable matching.

Rotation exposed in Table T

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

$$f_T(A) = D, s_T(A) = F, \ell_T(A) = F$$
$$f_T(C) = F, s_T(C) = D, \ell_T(C) = D$$
$$f_T(D) = C, s_T(D) = A, \ell_T(D) = A$$
$$f_T(F) = A, s_T(F) = C, \ell_T(F) = C$$

Rotation exposed in Table T

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

Rotation $\rho = (x_0, y_0), \cdots, (x_{r-1}, y_{r-1})$

$y_i = f_T(x_i)$ and $y_{i+1} = s_T(x_i)$

for all $i = 0, \cdots, r-1$, $i+1$ is taken modulo r

(A,D), (C,F) is a rotation exposed in T

Phase 2

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

- Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.
- As a result, all successors of x_i in y_{i+1}'s reduced lists can be deleted

(A,D), (C,F) is a rotation exposed in T

Phase 2

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

- Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.
- As a result, all successors of x_i in y_{i+1}'s reduced lists can be deleted

(A,D), (C,F) is a rotation exposed in T

A	F
C	D
A	

Phase 2

A		D	F		
B	E				
C		F			D
D	C	A			
E	B				
F	A			C	

- Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.
- As a result, all successors of x_i in y_{i+1}'s reduced lists can be deleted

(A,D), (C,F) is a rotation exposed in T

A	F
C	D
A	

Phase 2

A			F		
B	E				
C					D
D	C				
E	B				
F	A				

- Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.
- As a result, all successors of x_i in y_{i+1}'s reduced lists can be deleted

(A,D), (C,F) is a rotation exposed in T

A	F
C	D
A	

Phase 2

A	F
B	E
C	D

A	B	D	F	E	C
B	E	C	F	D	A
C	B	F	E	A	D
D	C	A	F	B	E
E	B	A	F	C	D
F	A	D	E	C	B

Irving's Stable Roommates Algorithm

```
assign each person to be free ;
while some free person x has a nonempty list do
begin
    y := first person on x's list ; {x proposes to y}
    if some person z is semiengaged to y then
        assign z to be free ; {y rejects z}
    assign x to be semiengaged to y ;
    for each successor x' of x on y's list do
        delete the pair {x', y} from the preference table
end
```

```
T := phase-1 table ;
while (some list in T has more than one entry)
                and (no list in T is empty) do
begin
    find a rotation ρ exposed in T ;
    T := T / ρ { eliminate ρ }
end ;
if some list in T is empty then
    report instance unsolvable
else
    output T, which is a stable matching
```

Running time $O(n^2)$

Correctness

- We have preserved at least 1 stable matching.
- Phase 1: Pairs eliminated are not part of any stable matching
- Phase 2: If input has a stable matching, then there exists a stable matching after rotations eliminations

Phase 1

First phase consisting of a sequence of “proposals”

Step 1: Proposal: 1. If x receives a proposal from y, then:

(a) x rejects it at once if x already holds a better proposal;

(b) x holds it for consideration otherwise, x rejects any poorer proposal x currently hold.

2. An individual x proposes to the others in the order in which they appear in x’s preference list, stopping when a promise of consideration is received; any subsequent rejection causes x to continue immediately with the sequence of proposals

Step 2: Short List: If *q* holds a proposal from *p*, then remove from *q*'s list all participants *x* after *p*, and symmetrically, for each removed participant *x*, we remove *q* from *x*'s list

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Proof: Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first.
- Suppose for a contradiction that M be a stable matching where x and y are partners:

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Proof: Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first.
- Suppose for a contradiction that M be a stable matching where x and y are partners:
  - y rejects x- y held (received) a better proposal, say z

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Proof: Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first.
- Suppose for a contradiction that M be a stable matching where x and y are partners:
  - y rejects x- y held (received) a better proposal, say z
  - z prefers partner of z (say w) in M over y. (else (y,z) is a blocking pair)

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Proof: Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first.
- Suppose for a contradiction that M be a stable matching where x and y are partners:
  - y rejects x- y held (received) a better proposal, say z
  - z prefers partner of z (say w) in M over y. (else (y,z) is a blocking pair)
  - w must have rejected z, therefore w proposed to y (this rejection happened before y rejects x)

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Proof: Suppose that of all the rejections involving two participants who are partners in a stable matching, the rejection of x by y is chronologically first.
- Suppose for a contradiction that M be a stable matching where x and y are partners:
  - y rejects x- y held (received) a better proposal, say z
  - z prefers partner of z (say w) in M over y. (else (y,z) is a blocking pair)
  - w must have rejected z, therefore w proposed to y (this rejection happened before y rejects x)
  - (z,w) is matched in M and rejected pair in phase 1 before (x,y) pair. A contradiction.

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Corollary 1: If, at any stage of the proposal process, x proposes to y, then in a stable matching:
  (1) x cannot have a better partner than y
  (2) y cannot have a worse partner than x

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Corollary 1: If, at any stage of the proposal process, x proposes to y, then in a stable matching:
  (1) x cannot have a better partner than y
  (2) y cannot have a worse partner than x
- Proof: (1) x proposed to y as it was rejected by everyone that x prefers over y

  (2) If y and z are partners in a stable matching. y prefers x over z and by (1) x prefers y over its matched partner. (x,y) is a blocking pair

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Corollary 1: If, at any stage of the proposal process, x proposes to y, then in a stable matching:

  (1) x cannot have a better partner than y

  (2) y cannot have a worse partner than x
- Corollary 2: If every person holds a proposal, then reduced preference list obtained in Step 2 of phase 1 have following:
  - (1) y is first on x's list and x is last on y's
  - (2) b appears on a's list iff a appears on b's.

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Corollary 1: If, at any stage of the proposal process, x proposes to y, then in a stable matching:

  (1) x cannot have a better partner than y

  (2) y cannot have a worse partner than x
- Corollary 2: If every person holds a proposal, then reduced preference list obtained in Step 2 of phase 1 have following:
  - (1) y is first on x's list and x is last on y's
  - (2) b appears on a's list iff a appears on b's.
- If in the reduced preference lists, every list contains one person, then the lists specify a stable matching.

Correctness: Phase 1

- Lemma 1: If y rejects x in the proposal sequence in Phase 1, then x and y cannot be partners in a stable matching.
- Corollary 1: If, at any stage of the proposal process, x proposes to y, then in a stable matching:

  (1) x cannot have a better partner than y

  (2) y cannot have a worse partner than x
- Corollary 2: If every person holds a proposal, then reduced preference list obtained in Step 2 of phase 1 have following:
  - (1) **y is first on x's list and x is last on y's**
  - (2) b appears on a's list iff a appears on b's.
- If in the reduced preference lists, every list contains one person, then the lists specify a stable matching.

Phase 2

- Eliminate Rotations and reduce table size
- Iterative process that will result in either one person running out of people to propose to (in which case no stable matching exists), or every preference list shrinking to a single person, in which case they specify a stable matching.

Correctness: Phase 2

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

Rotation $\rho = (x_0, y_0), \cdots, (x_{r-1}, y_{r-1})$

$y_i = f_T(x_i)$ and $y_{i+1} = s_T(x_i)$, we also know $x_i = \ell_T(y_i)$

for all $i = 0, \cdots, r-1$, $i+1$ is taken modulo r

Correctness: Phase 2

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

Rotation $\rho = (x_0, y_0), \cdots, (x_{r-1}, y_{r-1})$

$y_i = f_T(x_i)$ and $y_{i+1} = s_T(x_i)$, we also know $x_i = \ell_T(y_i)$

for all $i = 0, \cdots, r-1, i+1$ is taken modulo r

Rotation Elimination: Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.

Correctness: Phase 2

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

Rotation $\rho = (x_0, y_0), \cdots, (x_{r-1}, y_{r-1})$

$y_i = f_T(x_i)$ and $y_{i+1} = s_T(x_i)$, we also know $x_i = \ell_T(y_i)$

for all $i = 0, \cdots, r-1, i+1$ is taken modulo r

Rotation Elimination: Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.

y_0 *rejects* x_0 $---$ x_0 *proposes to* y_1
y_1 *rejects* x_1 $---$ x_1 *proposes to* y_2
y_2 *rejects* x_2 $---$ x_2 *proposes to* y_3
…
y_{r-1} *rejects* x_{r-1} $---$ x_{r-1} *proposes to* y_0

Correctness: Phase 2

$f_T(x), s_T(x)$, and $\ell_T(x)$ represent the first, second, and last entries in T

Rotation $\rho = (x_0, y_0), \cdots, (x_{r-1}, y_{r-1})$

$y_i = f_T(x_i)$ and $y_{i+1} = s_T(x_i)$, we also know $x_i = \ell_T(y_i)$

for all $i = 0, \cdots, r-1, i+1$ is taken modulo r

Rotation Elimination: Force each y_i to reject the proposal held from x_i, forcing each x_i to propose to y_{i+1}, the second person in the reduced list.

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i,

(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i,

(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i,

Proof: (1) x_i and y_i are partners in a stable matching contained in the reduced lists.

- Since y_i is second in x_{i-1}'s it follows that x_{i-1} is at least present on y_i's reduced list.
- Additionally, since x_i is last on y_i's reduced list, it follows that y_i prefers x_{i-1} to x_i .
- Then for stability, x_{i-1} must be partnered by someone he prefers to y_i , and the only qualifying participant in x_{i-1}'s list is y_{i-1} as $f_T(x_{i-1}) = y_{i-1}$.
- Repeating this argument shows that x_i and y_i must be partners for all values of i

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i,

(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Correctness: Phase 2

Lemma 2:(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Proof: (2) M: x_i and y_i are partners for all i.

Correctness: Phase 2

Lemma 2:(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Proof: (2) M: x_i and y_i are partners for all i.

M': matching in which each x_i is partnered by $y_{i+1}(s_T(x_i))$,

If participants not present in X-set or Y-set in rotation, then match same as M .

Correctness: Phase 2

Lemma 2:(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Proof: (2) M: x_i and y_i are partners for all i.

M': matching in which each x_i is partnered by $y_{i+1}(s_T(x_i))$,

If participants not present in X-set or Y-set in rotation, then match same as M .

Claim: M' is stable.

Correctness: Phase 2

Lemma 2:(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Proof: (2) M: x_i and y_i are partners for all i.

M': matching in which each x_i is partnered by $y_{i+1}(s_T(x_i))$,

If participants not present in X-set or Y-set in rotation, then match same as M .

Claim: M' is stable

Each participant of Y finds a better partner in M' (will not create a blocking pair).

The only individuals who do worse than M are the members of X.

Correctness: Phase 2

Lemma 2:(2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Proof: (2) M: x_i and y_i are partners for all i.

M’: matching in which each x_i is partnered by $y_{i+1}(s_T(x_i))$,

If participants not present in X-set or Y-set in rotation, then match same as M .

Claim: M’ is stable

Each participant of Y finds a better partner in M’ (will not create a blocking pair).

The only individuals who do worse than M are the members of X.

If x_i prefers z over y_{i+1}: $(\boldsymbol{x_i}, \boldsymbol{z})$**- partners in M**-- z gets better partner in M’

$\boldsymbol{x_i}$ **prefers** $\boldsymbol{z}$ **to** $\boldsymbol{y_i}$ -- since $y_i = f_T(x_i)$, z is not in reduced list of x_i

$\boldsymbol{z}$ **is ranked between** $\boldsymbol{y_i}$ **and** $\boldsymbol{y_{i+1}}$ **in original list--** z is not in reduced list of x_i

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i, (2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i, (2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Corollary 1: If the original problem instance allows for a stable matching, then there is a stable matching contained in a completely reduced set of preference lists.

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i, (2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Corollary 1: If the original problem instance allows for a stable matching, then there is a stable matching contained in a completely reduced set of preference lists.

Corollary 2: If one or more among a reduced set of preference lists is empty, then the original problem instance admits no stable matching.

Correctness: Phase 2

Lemma 2: (1) In a stable matching contained in reduced lists, either x_i and y_i are partners for all values of i or for no value of i, (2) If there is such a stable matching in which x_i and y_i are partners, then there is another in which they are not.

Corollary 1: If the original problem instance allows for a stable matching, then there is a stable matching contained in a completely reduced set of preference lists.

Corollary 2: If one or more among a reduced set of preference lists is empty, then the original problem instance admits no stable matching

Lemma 3. If in a completely reduced set of preference lists, every list contains just one person, then the lists specify a stable matching.

Thank You!