Today.

Comment: Add 0.

Add \((k - k)\).

Induction: Some quibbles.

Induction and Recursion

Couple of more induction proofs.

Stable Marriage.
Today.

Comment: Add 0.
Comment: Add 0.
Add \((k - k)\).
Comment: Add 0.

Add \((k - k)\).

Induction: Some quibbles.
Comment: Add 0.

Add \((k - k)\).

Induction: Some quibbles.

Induction and Recursion
Comment: Add 0.
Add \((k - k)\).
Induction: Some quibbles.
Induction and Recursion
Couple of more induction proofs.
Comment: Add 0.

Add \((k - k)\).

Induction: Some quibbles.

Induction and Recursion

Couple of more induction proofs.

Stable Marriage.
Some quibbles.

The induction principle works on the natural numbers.
The induction principle works on the natural numbers. Proves statements of form: $\forall n \in \mathbb{N}, P(n)$. 

Base Case: typically start at 3. Since $\forall n \in \mathbb{N}, Q(n) \Rightarrow Q(n+1)$ is trivially true before 3. 

Can you do induction over other things? Yes. Any set where any subset of the set has a smallest element. In some sense, the natural numbers.
Some quibbles.

The induction principle works on the natural numbers.
Proves statements of form: $\forall n \in \mathbb{N}, P(n)$.
Yes.
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: $\forall n \in \mathbb{N}, P(n)$.

Yes.

What if the statement is only for $n \geq 3$?
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: $\forall n \in \mathbb{N}, P(n)$.

Yes.

What if the statement is only for $n \geq 3$?

$$\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)$$
Some quibbles.

The induction principle works on the natural numbers.
Proves statements of form: \(\forall n \in \mathbb{N}, P(n)\).

Yes.

What if the statement is only for \(n \geq 3\)?

\[
\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)
\]

Restate as:
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: \( \forall n \in \mathbb{N}, P(n) \).

Yes.

What if the statement is only for \( n \geq 3 \)?

\[
\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)
\]

Restate as:

\[
\forall n \in \mathbb{N}, Q(n) \text{ where } Q(n) = "(n \geq 3) \implies P(n)".
\]
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: \( \forall n \in \mathbb{N}, P(n) \).

Yes.

What if the statement is only for \( n \geq 3 \)?

\[
\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)
\]

Restate as:

\[
\forall n \in \mathbb{N}, Q(n) \text{ where } Q(n) = "(n \geq 3) \implies P(n)".
\]

Base Case: typically start at 3.
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: \( \forall n \in \mathbb{N}, P(n) \).

Yes.

What if the statement is only for \( n \geq 3 \)?

\[
\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)
\]

Restate as:

\[
\forall n \in \mathbb{N}, Q(n) \text{ where } Q(n) = "(n \geq 3) \implies P(n)".
\]

Base Case: typically start at 3.

Since \( \forall n \in \mathbb{N}, Q(n) \implies Q(n+1) \) is trivially true before 3.
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: \( \forall n \in \mathbb{N}, P(n) \).

Yes.

What if the statement is only for \( n \geq 3 \)?

\[
\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)
\]

Restate as:

\[
\forall n \in \mathbb{N}, Q(n) \text{ where } Q(n) = "(n \geq 3) \implies P(n)".
\]

Base Case: typically start at 3.

Since \( \forall n \in \mathbb{N}, Q(n) \implies Q(n+1) \) is trivially true before 3.

Can you do induction over other things? Yes.
Some quibbles.

The induction principle works on the natural numbers.
Proves statements of form: $\forall n \in \mathbb{N}, P(n)$.

Yes.

What if the statement is only for $n \geq 3$?

$$\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)$$

Restate as:

$$\forall n \in \mathbb{N}, Q(n) \text{ where } Q(n) = "(n \geq 3) \implies P(n)".$$  

Base Case: typically start at 3.

Since $\forall n \in \mathbb{N}, Q(n) \implies Q(n+1)$ is trivially true before 3.

Can you do induction over other things? Yes.

Any set where any subset of the set has a smallest element.
Some quibbles.

The induction principle works on the natural numbers. Proves statements of form: $\forall n \in \mathbb{N}, P(n)$.

Yes.

What if the statement is only for $n \geq 3$?

$\forall n \in \mathbb{N}, (n \geq 3) \implies P(n)$

Restate as:

$\forall n \in \mathbb{N}, Q(n)$ where $Q(n) = "(n \geq 3) \implies P(n)"$.

Base Case: typically start at 3.

Since $\forall n \in \mathbb{N}, Q(n) \implies Q(n+1)$ is trivially true before 3.

Can you do induction over other things? Yes.

Any set where any subset of the set has a smallest element.

In some sense, the natural numbers.
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$. 
Strong Induction and Recursion.

Thm: For every natural number \( n \geq 12 \), \( n = 4x + 5y \).

Instead of proof, let's write some code!

```python
def find_x_y(n):
    if (n == 12):
        return (3, 0)
    elif (n == 13):
        return (2, 1)
    elif (n == 14):
        return (1, 2)
    elif (n == 15):
        return (0, 3)
    else:
        (x_prime, y_prime) = find_x_y(n - 4)
        return (x_prime + 1, y_prime)

# Base cases:
P(12), P(13), P(14), P(15).

# Strong Induction step:
Recursive call is correct:
P(n - 4) = \Rightarrow P(n).

n - 4 = 4x' + 5y' = \Rightarrow n = 4(x' + 1) + 5y'.

Slight differences: showed for all \( n \geq 16 \) that \( n - 1 = 4P(i) = \Rightarrow P(n) \).
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

def find-x-y(n):
    if (n==12): return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return(x'+1,y')
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let's write some code!

```python
def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return(x'+1,y')
```

Base cases:
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let's write some code!

```python
def find_x_y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find_x_y(n-4)
        return (x'+1,y')
```

Base cases: $P(12)$
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.
Instead of proof, let's write some code!

```python
def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x’,y’) = find-x-y(n-4)
        return (x’+1,y’)
```

Base cases: $P(12)$, $P(13)$
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let's write some code!

```python
def find_x_y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find_x_y(n-4)
        return(x'+1,y')
```

Base cases: $P(12)$, $P(13)$, $P(14)$
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return (x'+1,y')

Base cases: $P(12)$, $P(13)$, $P(14)$, $P(15)$. 
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x’,y’) = find-x-y(n-4)
        return(x’+1,y’)

Base cases: P(12), P(13), P(14), P(15). Yes.
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return(x'+1,y')


Strong Induction step:
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x’,y’) = find-x-y(n-4)
        return (x’+1,y’)


Strong Induction step:
Recursive call is correct: $P(n–4)$
Strong Induction and Recursion.

Thm: For every natural number \( n \geq 12 \), \( n = 4x + 5y \).

Instead of proof, let’s write some code!

```python
def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x’,y’) = find-x-y(n-4)
        return(x’+1,y’)
```

Base cases: \( P(12) \), \( P(13) \), \( P(14) \), \( P(15) \). Yes.

Strong Induction step:
   Recursive call is correct: \( P(n – 4) \implies P(n) \).
Strong Induction and Recursion.

Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let's write some code!

```python
def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return(x'+1,y')
```

Base cases: $P(12), P(13), P(14), P(15)$. Yes.

Strong Induction step:
Recursive call is correct: $P(n-4) \implies P(n)$.

$n - 4 = 4x' + 5y' \implies n = 4(x' + 1) + 5(y')$
Strong Induction and Recursion.

Thm: For every natural number \( n \geq 12 \), \( n = 4x + 5y \).

Instead of proof, let’s write some code!

```python
def find-x-y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find-x-y(n-4)
        return(x'+1,y')
```

Base cases: \( P(12) \), \( P(13) \), \( P(14) \), \( P(15) \). Yes.

Strong Induction step:

Recursive call is correct: \( P(n-4) \implies P(n) \).
\[
    n - 4 = 4x' + 5y' \implies n = 4(x' + 1) + 5(y')
\]
Thm: For every natural number $n \geq 12$, $n = 4x + 5y$.

Instead of proof, let’s write some code!

```python
def find_x_y(n):
    if (n==12) return (3,0)
    elif (n==13): return(2,1)
    elif (n==14): return(1,2)
    elif (n==15): return(0,3)
    else:
        (x',y') = find_x_y(n-4)
        return(x'+1,y')
```

Base cases: $P(12), P(13), P(14), P(15)$. Yes.

Strong Induction step:
    Recursive call is correct: $P(n-4) \implies P(n)$.
    $n - 4 = 4x' + 5y' \implies n = 4(x' + 1) + 5(y')$

Slight differences: showed for all $n \geq 16$ that $\land_{i=4}^{n-1} P(i) \implies P(n)$. 

Strong Induction and Recursion.
Strengthening: need to...

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$.

Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

$\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \leq 2 + \frac{1}{(k+1)^2}$.

Uh oh?

Hmmm...

It better be that any sum is strictly less than $2$.

How much less?

At least by $1$ for $S_k$.

"$S_k \leq 2 - 1 \cdot (k+1)$" $\Rightarrow$ "$S_{k+1} \leq 2$".

Induction step works!

No!

Not the same statement!!!!

Need to prove "$S_{k+1} \leq 2 - 1 \cdot (k+2)$".
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Base: \( P(1) \).
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$. 
Strengthening: need to...

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.

Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$. 

Uh oh?

Hmmm...

It better be that any sum is strictly less than 2.

How much less?

At least by $(k+1)^2$ for $S_k$.

"$S_k \leq 2 - (k+2)^2$" =⇒ "$S_{k+1} \leq 2$".

Induction step works!

No!

Not the same statement!!!!

Need to prove "$S_{k+1} \leq 2 - (k+2)^2$".

Darn!!!
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.
Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} \]

Uh oh?
Hmmm...
It better be that any sum is strictly less than $2$.
How much less?
At least by $(k + 1)^2$ for $S_k$.

$S_k \leq 2 - 1 \left( k + 1 \right)^2$ \[ \Rightarrow \]

$S_{k+1} \leq 2$.

Induction step works!
No!
Not the same statement!!!!
Need to prove $S_{k+1} \leq 2 - 1 \left( k + 2 \right)^2$. 

Darn!!!
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]
Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]

\[
\leq 2 + \frac{1}{(k+1)^2}.
\]
Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\begin{align*}
\sum_{i=1}^{k+1} \frac{1}{i^2} &= \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \\
&\leq 2 + \frac{1}{(k+1)^2}.
\end{align*}
\]

Uh oh?
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \)).

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]

\[
\leq 2 + \frac{1}{(k+1)^2}
\]

Uh oh?

Hmmm...
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.

Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

$\sum_{i=1}^{k+1} \frac{1}{i^2}$

$= \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}$.

$\leq 2 + \frac{1}{(k+1)^2}$.

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.

Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

$\sum_{i=1}^{k+1} \frac{1}{i^2}$

$= \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \cdot$

$\leq 2 + \frac{1}{(k+1)^2}$

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.

How much less?
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.

Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \leq 2 + \frac{1}{(k+1)^2}
\]

Uh oh?

Hmmm... it better be that any sum is strictly less than 2.

How much less? At least by $\frac{1}{(k+1)^2}$ for $S_k$. 
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Base: \( P(1) \) \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\begin{align*}
\sum_{i=1}^{k+1} \frac{1}{i^2} &= \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \\
&\leq 2 + \frac{1}{(k+1)^2}
\end{align*}
\]

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.

How much less? At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)”
**Theorem:** For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}.$)

**Base:** $P(1)$. $1 \leq 2$.

**Ind Step:** $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \leq 2 + \frac{1}{(k+1)^2}
\]

Uh oh?

Hmmm... It better be that any sum is *strictly less than* 2.

**How much less?** At least by $\frac{1}{(k+1)^2}$ for $S_k$.

"$S_k \leq 2 - \frac{1}{(k+1)^2}$" $\implies$ "$S_{k+1} \leq 2$"
Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). \( (S_n = \sum_{i=1}^{n} \frac{1}{i^2}.) \)

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]
\[
\leq 2 + \frac{1}{(k+1)^2}
\]

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.

How much less? At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)” \( \implies \) “\( S_{k+1} \leq 2 \)”

Induction step works!
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). \((S_n = \sum_{i=1}^{n} \frac{1}{i^2}.)\)

Base: \( P(1) \). \( 1 \leq 2 \).
Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[ \sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \leq 2 + \frac{1}{(k+1)^2} \]

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.

How much less? At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)” \( \implies \) “\( S_{k+1} \leq 2 \)”

Induction step works! No!
Strengthening: need to...

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Base: $P(1)$. $1 \leq 2$.
Ind Step: $\sum_{i=1}^{k} \frac{1}{i^2} \leq 2$.

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]

\[
\leq 2 + \frac{1}{(k+1)^2}.
\]

Uh oh?

Hmmm... It better be that any sum is strictly less than 2.

How much less? At least by $\frac{1}{(k+1)^2}$ for $S_k$.

“$S_k \leq 2 - \frac{1}{(k+1)^2}$” $\implies$ “$S_{k+1} \leq 2$”

Induction step works! No! Not the same statement!!!!
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2} \leq 2 + \frac{1}{(k+1)^2}
\]

Uh oh?

Hmmm... It better be that any sum is \textit{strictly less than} 2.

How much less? At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)” \( \implies \) “\( S_{k+1} \leq 2 \)”

Induction step works! No! Not the same statement!!!!

Need to prove “\( S_{k+1} \leq 2 - \frac{1}{(k+2)^2} \).”
Strengthening: need to...

Theorem: For all \( n \geq 1, \sum_{i=1}^{n} \frac{1}{i^2} \leq 2. \) \((S_n = \sum_{i=1}^{n} \frac{1}{i^2}\).

Base: \( P(1). \) \( 1 \leq 2. \)

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2. \)

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]

\[ \leq 2 + \frac{1}{(k+1)^2} \]

Uh oh?

Hmmm... It better be that any sum is \textit{strictly less than} 2.

How much less? At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)” \( \implies \) “\( S_{k+1} \leq 2 \)”

Induction step works! No! Not the same statement!!!!

Need to prove “\( S_{k+1} \leq 2 - \frac{1}{(k+2)^2} \).”
Strengthening: need to...

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 \). \( (S_n = \sum_{i=1}^{n} \frac{1}{i^2}.) \)

Base: \( P(1) \). \( 1 \leq 2 \).

Ind Step: \( \sum_{i=1}^{k} \frac{1}{i^2} \leq 2 \).

\[
\sum_{i=1}^{k+1} \frac{1}{i^2} = \sum_{i=1}^{k} \frac{1}{i^2} + \frac{1}{(k+1)^2}.
\]

\[
\leq 2 + \frac{1}{(k+1)^2}.
\]

Uh oh?

Hmmm...  It better be that any sum is strictly less than 2.

How much less?  At least by \( \frac{1}{(k+1)^2} \) for \( S_k \).

“\( S_k \leq 2 - \frac{1}{(k+1)^2} \)” \( \implies \) “\( S_{k+1} \leq 2 \)”

Induction step works!  No!  Not the same statement!!!!

Need to prove “\( S_{k+1} \leq 2 - \frac{1}{(k+2)^2} \).”

Darn!!!
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Strengthening: how?
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)
Proof:
Ind hyp: $P(k)$

Can you? Subtracting off a quadratically decreasing function every time.
Maybe a linearly decreasing function to keep positive?
Try $f(k) = \frac{1}{k(k+1)} \leq \frac{1}{k} - \frac{1}{k+1}$?

$1 \leq k + 1$. Multiplied by $k + 1$. So yes!
Strengthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Proof:
Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$
Prove: $P(k + 1)$
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — "S_k \leq 2 - f(k)"
Prove: $P(k + 1) – "S_{k+1} \leq 2 - f(k + 1)"$
**Strengthening: how?**

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

**Proof:**

Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$

Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) \rightarrow "S_k \leq 2 - f(k)"
Prove: $P(k+1) \rightarrow "S_{k+1} \leq 2 - f(k+1)"

S(k + 1) = S_k + \frac{1}{(k+1)^2}$
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:

Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$

Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$

$S(k + 1) = S_k + \frac{1}{(k+1)^2}$

$\leq 2 - f(k) + \frac{1}{(k+1)^2}$
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$
Prove: $P(k+1) — “S_{k+1} \leq 2 - f(k + 1)”$

\[ S(k + 1) = S_k + \frac{1}{(k+1)^2} \]
\[ \leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.} \]
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — " S_k \leq 2 - f(k)"
Prove: $P(k+1) — " S_{k+1} \leq 2 - f(k+1)"

$$S(k+1) = S_k + \frac{1}{(k+1)^2}$$

$\leq 2 - f(k) + \frac{1}{(k+1)^2}$ By ind. hyp.
Strengthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Proof:
Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”

Prove: \( P(k + 1) \) — “\( S_{k+1} \leq 2 - f(k + 1) \)”

\[
S(k + 1) = S_k + \frac{1}{(k+1)^2}
\]

\[
\leq 2 - f(k) + \frac{1}{(k+1)^2} \quad \text{By ind. hyp.}
\]

Choose \( f(k + 1) \leq f(k) - \frac{1}{(k+1)^2} \).
Strenthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Proof:
Ind hyp: \( P(k) \) — "\( S_k \leq 2 - f(k) \)"
Prove: \( P(k+1) \) — "\( S_{k+1} \leq 2 - f(k + 1) \)"

\[
S(k + 1) = S_k + \frac{1}{(k+1)^2} \leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}
\]

Choose \( f(k + 1) \leq f(k) - \frac{1}{(k+1)^2} \).
\[\implies S(k + 1) \leq 2 - f(k + 1).\]
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$.

$(S_n = \sum_{i=1}^{n} \frac{1}{i^2})$.

Proof:

Ind hyp: $P(k)$ — “$S_k \leq 2 - f(k)$”

Prove: $P(k + 1)$ — “$S_{k+1} \leq 2 - f(k + 1)$”

$S(k + 1) = S_k + \frac{1}{(k+1)^2}$

$\leq 2 - f(k) + \frac{1}{(k+1)^2}$ By ind. hyp.

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.  

$\implies S(k + 1) \leq 2 - f(k + 1)$.

Can you?
Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$
Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$

$$S(k + 1) = S_k + \frac{1}{(k+1)^2}$$

$$\leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}$$

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$. 

$$\implies S(k + 1) \leq 2 - f(k + 1).$$

Can you?
**Strengthening: how?**

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

**Proof:**

Ind hyp: $P(k) \rightarrow "S_k \leq 2 - f(k)"

Prove: $P(k + 1) \rightarrow "S_{k+1} \leq 2 - f(k + 1)"

$S(k + 1) = S_k + \frac{1}{(k+1)^2}

\leq 2 - f(k) + \frac{1}{(k+1)^2}$ By ind. hyp.

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.

$$\implies S(k + 1) \leq 2 - f(k + 1).$$

Can you?

Subtracting off a quadratically decreasing function every time.
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$
Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$

$$S(k + 1) = S_k + \frac{1}{(k+1)^2} \leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}$$

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.  
$$\implies S(k + 1) \leq 2 - f(k + 1).$$

Can you?
Subtracting off a quadratically decreasing function every time. Maybe a linearly decreasing function to keep positive?
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:
Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$
Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$

$S(k + 1) = S_k + \frac{1}{(k+1)^2}$

$\leq 2 - f(k) + \frac{1}{(k+1)^2}$ By ind. hyp.

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.

$\implies S(k + 1) \leq 2 - f(k + 1)$.

Can you?
  Subtracting off a quadratically decreasing function every time.
  Maybe a linearly decreasing function to keep positive?
Try $f(k) = \frac{1}{k}$
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:

Ind hyp: $P(k) \Rightarrow "S_k \leq 2 - f(k)"

Prove: $P(k+1) \Rightarrow "S_{k+1} \leq 2 - f(k+1)"

\[
S(k + 1) = S_k + \frac{1}{(k+1)^2} \\
\leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}
\]

Choose $f(k+1) \leq f(k) - \frac{1}{(k+1)^2}$. 

$\Rightarrow S(k + 1) \leq 2 - f(k + 1)$.

Can you?

Subtracting off a quadratically decreasing function every time. 
Maybe a linearly decreasing function to keep positive?

Try $f(k) = \frac{1}{k}$

$\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2}$?
Strengthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

Proof:
Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”
Prove: \( P(k+1) \) — “\( S_{k+1} \leq 2 - f(k+1) \)”

\[
S(k+1) = S_k + \frac{1}{(k+1)^2}
\leq 2 - f(k) + \frac{1}{(k+1)^2} \quad \text{By ind. hyp.}
\]

Choose \( f(k+1) \leq f(k) - \frac{1}{(k+1)^2} \).

\[
\implies S(k+1) \leq 2 - f(k+1).
\]

Can you?
Subtracting off a quadratically decreasing function every time.
Maybe a linearly decreasing function to keep positive?

Try \( f(k) = \frac{1}{k} \)

\[
\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} ?
\]
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:

Ind hyp: $P(k) — “S_k \leq 2 - f(k)”$

Prove: $P(k + 1) — “S_{k+1} \leq 2 - f(k + 1)”$

$$S(k + 1) = S_k + \frac{1}{(k+1)^2}$$
$$\leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}$$

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.

$$\implies S(k + 1) \leq 2 - f(k + 1).$$

Can you?

Subtracting off a quadratically decreasing function every time.
Maybe a linearly decreasing function to keep positive?

Try $f(k) = \frac{1}{k}$

$$\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} ?$$

$$1 \leq \frac{k+1}{k} - \frac{1}{k+1}$$
Strengthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). \( (S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).

Proof:
Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”
Prove: \( P(k+1) \) — “\( S_{k+1} \leq 2 - f(k+1) \)”

\[
S(k+1) = S_k + \frac{1}{(k+1)^2} \\
\leq 2 - f(k) + \frac{1}{(k+1)^2} \quad \text{By ind. hyp.}
\]

Choose \( f(k+1) \leq f(k) - \frac{1}{(k+1)^2} \).

\[ \implies S(k+1) \leq 2 - f(k+1). \]

Can you?
Subtracting off a quadratically decreasing function every time. 
Maybe a linearly decreasing function to keep positive?
Try \( f(k) = \frac{1}{k} \)

\[
\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} \quad ?
\]

\[ 1 \leq \frac{k+1}{k} - \frac{1}{k+1} \quad \text{Multiplied by } k + 1.
\]

\[ 1 \leq 1 + \left( \frac{1}{k} - \frac{1}{k+1} \right) \]
Strenthening: how?

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). \( (S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).

Proof:
Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”
Prove: \( P(k+1) \) — “\( S_{k+1} \leq 2 - f(k+1) \)”

\[
S(k+1) = S_k + \frac{1}{(k+1)^2} \\
\leq 2 - f(k) + \frac{1}{(k+1)^2} \quad \text{By ind. hyp.}
\]

Choose \( f(k+1) \leq f(k) - \frac{1}{(k+1)^2} \).

\[
\implies S(k+1) \leq 2 - f(k+1).
\]

Can you?
Subtracting off a quadratically decreasing function every time.
Maybe a linearly decreasing function to keep positive?
Try \( f(k) = \frac{1}{k} \)

\[
\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} \quad ?
\]

\[
1 \leq \frac{k+1}{k} - \frac{1}{k+1} \quad \text{Multiplied by } k + 1.
\]

\[
1 \leq 1 + \left( \frac{1}{k} - \frac{1}{k+1} \right) \quad \text{Some math.}
\]
**Strengthening: how?**

Theorem: For all \( n \geq 1 \), \( \sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n) \). (\( S_n = \sum_{i=1}^{n} \frac{1}{i^2} \).)

**Proof:**

Ind hyp: \( P(k) \) — “\( S_k \leq 2 - f(k) \)”

Prove: \( P(k+1) \) — “\( S_{k+1} \leq 2 - f(k+1) \)”

\[
S(k+1) = S_k + \frac{1}{(k+1)^2}
\]

\[
\leq 2 - f(k) + \frac{1}{(k+1)^2} \text{ By ind. hyp.}
\]

Choose \( f(k+1) \leq f(k) - \frac{1}{(k+1)^2} \).

\[
\implies S(k+1) \leq 2 - f(k+1).
\]

Can you?

Subtracting off a quadratically decreasing function every time. Maybe a linearly decreasing function to keep positive?

Try \( f(k) = \frac{1}{k} \)

\[
\frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} ?
\]

\[
1 \leq \frac{k+1}{k} - \frac{1}{k+1} \quad \text{Multiplied by } k + 1.
\]

\[
1 \leq 1 + \left( \frac{1}{k} - \frac{1}{k+1} \right) \quad \text{Some math. So yes!}
\]
Strengthening: how?

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - f(n)$. ($S_n = \sum_{i=1}^{n} \frac{1}{i^2}$.)

Proof:

Ind hyp: $P(k) \rightarrow "S_k \leq 2 - f(k)"

Prove: $P(k + 1) \rightarrow "S_{k+1} \leq 2 - f(k+1)"

\[ S(k + 1) = S_k + \frac{1}{(k+1)^2} \]
\[ \leq 2 - f(k) + \frac{1}{(k+1)^2} \]
By ind. hyp.

Choose $f(k + 1) \leq f(k) - \frac{1}{(k+1)^2}$.

\[ \Rightarrow S(k + 1) \leq 2 - f(k+1). \]

Can you?

Subtracting off a quadratically decreasing function every time.
Maybe a linearly decreasing function to keep positive?

Try $f(k) = \frac{1}{k}$

\[ \frac{1}{k+1} \leq \frac{1}{k} - \frac{1}{(k+1)^2} ? \]

1 \[ \leq \frac{k+1}{k} - \frac{1}{k+1} \]
Multiplied by $k + 1$.

1 \[ \leq 1 + \left( \frac{1}{k} - \frac{1}{k+1} \right) \]
Some math. So yes!

Theorem: For all $n \geq 1$, $\sum_{i=1}^{n} \frac{1}{i^2} \leq 2 - \frac{1}{n}$. 
Stable Marriage Problem

- Small town with \( n \) boys and \( n \) girls.
- Each girl has a ranked preference list of boys.
- Each boy has a ranked preference list of girls.

How should they be matched?
Stable Marriage Problem

- Small town with $n$ boys and $n$ girls.
Stable Marriage Problem

- Small town with $n$ boys and $n$ girls.
- Each girl has a ranked preference list of boys.
Stable Marriage Problem

- Small town with $n$ boys and $n$ girls.
- Each girl has a ranked preference list of boys.
- Each boy has a ranked preference list of girls.
Stable Marriage Problem

- Small town with $n$ boys and $n$ girls.
- Each girl has a ranked preference list of boys.
- Each boy has a ranked preference list of girls.

How should they be matched?
Count the ways..

- Maximize total satisfaction.
Count the ways..

- Maximize total satisfaction.
- Maximize number of first choices.
Count the ways..

- Maximize total satisfaction.
- Maximize number of first choices.
- Maximize worse off.
Count the ways..

- Maximize total satisfaction.
- Maximize number of first choices.
- Maximize worse off.
- Minimize difference between preference ranks.
The best laid plans..

Consider the couples..

- Jennifer and Brad
- Angelina and Billy-Bob
The best laid plans..

Consider the couples..

- Jennifer and Brad
- Angelina and Billy-Bob

Brad prefers Angelina to Jennifer.
The best laid plans..

Consider the couples..

- Jennifer and Brad
- Angelina and Billy-Bob

Brad prefers Angelina to Jennifer.
Angelina prefers Brad to BillyBob.
The best laid plans..

Consider the couples..

- Jennifer and Brad
- Angelina and Billy-Bob

Brad prefers Angelina to Jennifer.
Angelina prefers Brad to BillyBob.
Uh..oh.
So...

Produce a pairing where there is no running off!
So..

Produce a pairing where there is no running off!

**Definition:** A *pairing* is disjoint set of $n$ boy-girl pairs.
Produce a pairing where there is no running off!

**Definition:** A **pairing** is disjoint set of $n$ boy-girl pairs.

Example: A pairing $S = \{(Brad, Jen); (BillyBob, Angelina)\}$. 
Produce a pairing where there is no running off!

**Definition:** A **pairing** is disjoint set of $n$ boy-girl pairs.

Example: A pairing $S = \{(Brad, Jen); (BillyBob, Angelina)\}$.

**Definition:** A **rogue couple** $b, g^*$ for a pairing $S$: $b$ and $g^*$ prefer each other to their partners in $S$. 

So..
Produce a pairing where there is no running off!

**Definition:** A **pairing** is disjoint set of $n$ boy-girl pairs.

Example: A pairing $S = \{ (Brad, Jen); (BillyBob, Angelina) \}$.

**Definition:** A **rogue couple** $b, g^*$ for a pairing $S$: $b$ and $g^*$ prefer each other to their partners in $S$.

Example: Brad and Angelina are a rogue couple in $S$. 
A stable pairing??

Given a set of preferences.
A stable pairing??

Given a set of preferences.
Is there a stable pairing?
How does one find it?
A stable pairing??

Given a set of preferences.

Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

A | B   C   D
B | C   A   D
C | A   B   D
D | A   B   C

A ——— B

C ——— D
A stable pairing??

Given a set of preferences.
Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td></td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

![Diagram](image)
A stable pairing??

Given a set of preferences.

Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:

- A → C
- B → D
- C → A
- D → B
A stable pairing??

Given a set of preferences.
Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

A -> C -> B -> D
A stable pairing??

Given a set of preferences.

Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

\[
\begin{array}{c|cccc}
A & B & C & D \\
B & C & A & D \\
C & A & B & D \\
D & A & B & C \\
\end{array}
\]
A stable pairing??

Given a set of preferences.

Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

\[
\begin{array}{c|cccc}
A & B & C & D \\
B & C & A & D \\
C & A & B & D \\
D & A & B & C \\
\end{array}
\]
A stable pairing??

Given a set of preferences.

Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>C</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

A \_ B \_ C \_ D

C \_ D

A \_ B
A stable pairing??

Given a set of preferences.
Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>

Diagram:
- A is paired with B
- C is paired with D
- B is paired with C
- A is paired with C
A stable pairing??

Given a set of preferences.
Is there a stable pairing?
How does one find it?

Consider a single gender version: stable roommates.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>D</td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

A

B

C

D

A

B

C

D

C

D

A

B
The Traditional Marriage Algorithm.

Each Day:
1. Each boy proposes to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a string).
3. Rejected boy crosses rejecting girl off his list.

Stop when each girl gets exactly one proposal.

Does this terminate?...

produce a pairing?...

a stable pairing?

Do boys or girls do "better"?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy proposes to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a string.)
3. Rejected boy crosses rejecting girl off his list.

Stop when each girl gets exactly one proposal.

Does this terminate?

...produce a pairing?

....a stable pairing?

Do boys or girls do “better”?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy proposes to his favorite girl on his list.
The Traditional Marriage Algorithm.

Each Day:

1. Each boy **proposes** to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a **string**.)
The Traditional Marriage Algorithm.

Each Day:

1. Each boy **proposes** to his favorite girl on his list.

2. Each girl rejects all but her favorite proposer (whom she puts on a **string**.)

3. Rejected boy **crosses** rejecting girl off his list.
The Traditional Marriage Algorithm.

Each Day:

1. Each boy proposes to his favorite girl on his list.

2. Each girl rejects all but her favorite proposer (whom she puts on a string.)

3. Rejected boy crosses rejecting girl off his list.

Stop when each girl gets exactly one proposal.
The Traditional Marriage Algorithm.

Each Day:

1. Each boy \textit{proposes} to his favorite girl on his list.

2. Each girl rejects all but her favorite proposer (whom she puts on a \textit{string}.)

3. Rejected boy \textit{crosses} rejecting girl off his list.

Stop when each girl gets exactly one proposal.
Does this terminate?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy **proposes** to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a **string**.)
3. Rejected boy **crosses** rejecting girl off his list.

Stop when each girl gets exactly one proposal. Does this terminate?

...produce a pairing?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy **proposes** to his favorite girl on his list.

2. Each girl rejects all but her favorite proposer (whom she puts on a **string**.)

3. Rejected boy **crosses** rejecting girl off his list.

Stop when each girl gets exactly one proposal. Does this terminate?

...produce a pairing?

....a stable pairing?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy proposes to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a string.)
3. Rejected boy crosses rejecting girl off his list.

Stop when each girl gets exactly one proposal. Does this terminate?

...produce a pairing?

....a stable pairing?

Do boys or girls do “better”?
The Traditional Marriage Algorithm.

Each Day:

1. Each boy *proposes* to his favorite girl on his list.
2. Each girl rejects all but her favorite proposer (whom she puts on a *string*.)
3. Rejected boy *crosses* rejecting girl off his list.

Stop when each girl gets exactly one proposal. Does this terminate?

...produce a pairing?

....a stable pairing?

Do boys or girls do “better”? 

Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th></th>
<th>Girls</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 2 3</td>
<td></td>
<td>1 C A B</td>
</tr>
<tr>
<td>B</td>
<td>1 2 3</td>
<td></td>
<td>2 A B C</td>
</tr>
<tr>
<td>C</td>
<td>2 1 3</td>
<td></td>
<td>3 A C B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Girls</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 2 3</td>
<td></td>
<td>1 C A B</td>
</tr>
<tr>
<td>B</td>
<td>X 2 3</td>
<td></td>
<td>2 A B C</td>
</tr>
<tr>
<td>C</td>
<td>2 1 3</td>
<td></td>
<td>3 A C B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1  2  3</td>
<td></td>
<td>1  C  A  B</td>
</tr>
<tr>
<td>B</td>
<td>X  2  3</td>
<td></td>
<td>2  A  B  C</td>
</tr>
<tr>
<td>C</td>
<td>2  1  3</td>
<td></td>
<td>3  A  C  B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th>Boys</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Girls</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td></td>
<td>B, C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>1 2 3</td>
<td>1 C</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>X 2 3</td>
<td>A B</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>X 1 3</td>
<td>A C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>1 X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>A, B</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>C, A</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Days 1</th>
<th>Days 2</th>
<th>Days 3</th>
<th>Days 4</th>
<th>Days 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>A, C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B, C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>X</td>
<td>2</td>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>2</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>1</td>
<td>3</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>X, C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B, C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th></th>
<th></th>
<th>Girls</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>2</td>
<td>3</td>
<td></td>
<td>1</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>2</td>
<td>3</td>
<td></td>
<td>2</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>1</td>
<td>3</td>
<td></td>
<td>3</td>
<td>A</td>
<td>C</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>X, C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B, C</td>
<td>B</td>
<td>A, B</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>X</td>
<td>1</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>X</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>3</td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>3</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>X, C</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B, C</td>
<td>B</td>
<td>A, B</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th></th>
<th>Boys</th>
<th></th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>X</td>
<td>2 3</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>X</td>
<td>X 2</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>X 1</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>X, C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B</td>
<td>B</td>
<td>A, B</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>A, B</td>
<td>B</td>
<td></td>
<td>B</td>
</tr>
</tbody>
</table>
Example.

<table>
<thead>
<tr>
<th>Boys</th>
<th>Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A, B</td>
<td>A</td>
<td>X, C</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>B, C</td>
<td>B</td>
<td>A, B</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>
Termination.
Termination.

Every non-terminated day a boy crossed an item off the list.
Termination.

Every non-terminated day a boy crossed an item off the list.
Total size of lists?
Every non-terminated day a boy *crossed* an item off the list.
Total size of lists? $n$ boys, $n$ length list.
Termination.

Every non-terminated day a boy crossed an item off the list.
Total size of lists? $n$ boys, $n$ length list. $n^2$
Termination.

Every non-terminated day a boy crossed an item off the list.
Total size of lists? $n$ boys, $n$ length list. $n^2$
Terminates in at most $n^2 + 1$ steps!
It gets better every day for girls.

**Improvement Lemma:** It just gets better for girls.

If on day \( t \) a girl \( g \) has a boy \( b \) on a string, any boy, \( b' \), on \( g \)'s string for any day \( t' > t \) is at least as good as \( b \).

Let's apply lemma.

Girl “Alice” has boy “Bob” on string on day 5. She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

\( g \) - ‘Alice’, \( b \) - ‘Bob’, \( b' \) - ‘Jim’, \( t = 5 \), \( t' = 7 \).

Improvement Lemma says she prefers ‘Jim’.

On day 10, could Alice still have “Jim” on her string?

Yes. She likes her day 10 boy at least as much as her day 7 boy.

Here, \( b = b' \).

Why is lemma true?

Proof Idea: Because she can always keep the previous boy on the string.
It gets better every day for girls..

Improvement Lemma: It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t'$ > $t$ is at least as good as $b$.

Let's apply lemma. Girl "Alice" has boy "Bob" on string on day 5. She has boy "Jim" on string on day 7. Does Alice prefer "Jim" or "Bob"?

$g$ - 'Alice', $b$ - 'Bob', $b'$ - 'Jim', $t = 5$, $t' = 7$.

Improvement Lemma says she prefers 'Jim'.

On day 10, could Alice still have "Jim" on her string? Yes. She likes her day 10 boy at least as much as her day 7 boy.

Here, $b = b'$.

Why is lemma true?

Proof Idea: Because she can always keep the previous boy on the string.
It gets better every day for girls.

**Improvement Lemma:** It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string,
It gets better every day for girls..

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$.

Let’s apply lemma. Girl “Alice” has boy “Bob” on string on day 5. She has boy “Jim” on string on day 7. Does Alice prefer “Jim” or “Bob”?


Improvement Lemma says she prefers ‘Jim’.

On day 10, could Alice still have “Jim” on her string?

Yes. She likes her day 10 boy at least as much as her day 7 boy. Here, $b = b'$.

Why is lemma true?

**Proof Idea:** Because she can always keep the previous boy on the string.
It gets better every day for girls.

**Improvement Lemma:** It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$. 

Let's apply lemma.
Girl "Alice" has boy "Bob" on string on day 5.
She has boy "Jim" on string on day 7.
Does Alice prefer "Jim" or "Bob"?

$g$ - 'Alice',
$b$ - 'Bob',
$b'$ - 'Jim',
$t = 5,$
$t' = 7.$

Improvement Lemma says she prefers 'Jim'.

On day 10, could Alice still have "Jim" on her string?
Yes.
She likes her day 10 boy at least as much as her day 7 boy.
Here, $b = b'$. 

Why is lemma true?
Proof Idea:
Because she can always keep the previous boy on the string.
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Let's apply lemma.
Improvement Lemma: It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Let's apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.
It gets better every day for girls..

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.

She has boy “Jim” on string on day 7.
It gets better every day for girls.

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Let's apply lemma.

Girl “Alice” has boy “Bob” on string on day 5. She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”? 

On day 10, could Alice still have “Jim” on her string?
Yes. She likes her day 10 boy at least as much as her day 7 boy. Here, $b = b'$. Why is lemma true?
Proof Idea: Because she can always keep the previous boy on the string.
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5. She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$. 
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5. She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.
It gets better every day for girls.

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.

On day 10, could Alice still have “Jim” on her string?
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?  

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.
She likes her day 10 boy at least as much as her day 7 boy.  
Here, $b = b'$.
It gets better every day for girls..

**Improvement Lemma**: It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?


Improvement Lemma says she prefers ‘Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.

She likes her day 10 boy at least as much as her day 7 boy.
Here, $b = b'$.

Why is lemma true?
It gets better every day for girls..

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.

She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.

She likes her day 10 boy at least as much as her day 7 boy.

Here, $b = b'$.

Why is lemma true?
It gets better every day for girls.

**Improvement Lemma:** It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.

Does Alice prefer “Jim” or “Bob”?


Improvement Lemma says she prefers ‘Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.

She likes her day 10 boy at least as much as her day 7 boy.
Here, $b = b'$.

Why is lemma true?

Proof Idea:
It gets better every day for girls.

**Improvement Lemma:** It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Let’s apply lemma.

Girl “Alice” has boy “Bob” on string on day 5.
She has boy “Jim” on string on day 7.
Does Alice prefer “Jim” or “Bob”?

$g$ - ’Alice’, $b$ - ’Bob’, $b'$ - ’Jim’, $t = 5$, $t' = 7$.

Improvement Lemma says she prefers ’Jim’.

On day 10, could Alice still have “Jim” on her string? Yes.
She likes her day 10 boy at least as much as her day 7 boy.
Here, $b = b'$.

Why is lemma true?
Proof Idea: Because she can always keep the previous boy on the string.
Improvement Lemma

Improvement Lemma: It just gets better for girls.
Improvement Lemma

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$. 

Proof: 

$P(0)$ – true. Girl has $b$ on string. 

Assume $P(k)$. Let $b'$ be boy on string on day $t+k$. 

On day $t+k+1$, boy $b'$ comes back. Girl can choose $b'$, or do better with another boy, $b''$. That is, $b' \leq b$ by induction hypothesis. 

And $b''$ is better than $b'$ by algorithm. 

$\Rightarrow$ Girl does at least as well as with $b$. 

$P(k) = \Rightarrow P(k+1)$.

And by principle of induction.
Improvement Lemma

Improvement Lemma: It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Proof:
Improvement Lemma

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$ - - “boy on $g$’s string is at least as good as $b$ on day $t + k$”
Improvement Lemma

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$- “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$- true. Girl has $b$ on string.
Improvement Lemma

Improvement Lemma: It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Proof:
$P(k)$ - - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$ – true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$. 
Improvement Lemma: It just gets better for girls.

If on day \( t \) a girl \( g \) has a boy \( b \) on a string, any boy, \( b' \), on \( g \)'s string for any day \( t' > t \) is at least as good as \( b \).

Proof:

\( P(k) \)- - “boy on \( g \)'s string is at least as good as \( b \) on day \( t + k \)”

\( P(0) \)- true. Girl has \( b \) on string.

Assume \( P(k) \). Let \( b' \) be boy on string on day \( t + k \).

On day \( t + k + 1 \), boy \( b' \) comes back.
**Improvement Lemma**

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$ – “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$– true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy **on string** on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$,
Improvement Lemma

Improvement Lemma: It just gets better for girls.
If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Proof:
$P(k)$—“boy on $g$’s string is at least as good as $b$ on day $t + k$”
$P(0)$—true. Girl has $b$ on string.
Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.
On day $t + k + 1$, boy $b'$ comes back.
   Girl can choose $b'$, or do better with another boy, $b''$
Improvement Lemma

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$- “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$—true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is,
Improvement Lemma: It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$ - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$ – true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.
Improvement Lemma

Improvement Lemma: It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

Proof:
$P(k)$ - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$ – true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.

And $b''$ is better than $b'$ by algorithm.
Improvement Lemma

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$ - - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$– true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy **on string** on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

   Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.

   And $b''$ is better than $b'$ by algorithm.

   $\implies$ Girl does at least as well as with $b$. 
Improvement Lemma

Improvement Lemma: It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$'s string for any day $t' > t$ is at least as good as $b$.

Proof:
$P(k)$- - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$– true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.

And $b''$ is better than $b'$ by algorithm.

$\implies$ Girl does at least as well as with $b$.

$P(k) \implies P(k + 1)$. 
Improvement Lemma

**Improvement Lemma:** It just gets better for girls.

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$ - “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$ – true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.

And $b''$ is better than $b'$ by algorithm.

$\implies$ Girl does at least as well as with $b$.

$P(k) \implies P(k + 1)$. And by principle of induction.
Improvement Lemma

**Improvement Lemma: It just gets better for girls.**

If on day $t$ a girl $g$ has a boy $b$ on a string, any boy, $b'$, on $g$’s string for any day $t' > t$ is at least as good as $b$.

**Proof:**

$P(k)$- “boy on $g$’s string is at least as good as $b$ on day $t + k$”

$P(0)$—true. Girl has $b$ on string.

Assume $P(k)$. Let $b'$ be boy on string on day $t + k$.

On day $t + k + 1$, boy $b'$ comes back.

Girl can choose $b'$, or do better with another boy, $b''$

That is, $b' \leq b$ by induction hypothesis.

And $b''$ is better than $b'$ by algorithm.

$\implies$ Girl does at least as well as with $b$.

$P(k) \implies P(k + 1)$. And by principle of induction. \qed
Pairing when done.

**Lemma:** Every boy is matched at end.
Lemma: Every boy is matched at end.

Proof:
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.
**Lemma:** Every boy is matched at end.

**Proof:**
If not, a boy $b$ must have been rejected $n$ times.
Every girl has been proposed to by $b$, 
Pairing when done.

**Lemma:** Every boy is matched at end.

**Proof:**
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$,
and **Improvement lemma**
Pairing when done.

**Lemma:** Every boy is matched at end.

**Proof:**
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$, and **Improvement lemma**

$\implies$ each girl has a boy on a string.
**Lemma:** Every boy is matched at end.

**Proof:**
If not, a boy $b$ must have been rejected $n$ times.
Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.
Lemma: Every boy is matched at end.

Proof: If not, a boy $b$ must have been rejected $n$ times.
Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.
and each boy is on at most one string.
$n$ girls and $n$ boys.
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.
Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.
and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.
Lemma: Every boy is matched at end.

Proof: If not, a boy $b$ must have been rejected $n$ times. Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.

$\implies b$ must be on some girl’s string!
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.

$\implies b$ must be on some girl’s string!
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.

$\implies b$ must be on some girl’s string!

Contradiction.
Lemma: Every boy is matched at end.

Proof:
If not, a boy $b$ must have been rejected $n$ times.

Every girl has been proposed to by $b$, and Improvement lemma

$\implies$ each girl has a boy on a string.

and each boy is on at most one string.

$n$ girls and $n$ boys. Same number of each.

$\implies$ $b$ must be on some girl’s string!

Contradiction.
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
b^* & \quad \rightarrow \quad g^* \\
& \\
\end{align*}
\]

\[
\begin{align*}
b & \quad \rightarrow \quad g \\
& \\
\end{align*}
\]

Contradiction!
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
    b^* & \quad \text{-----} \quad g^* \\
    b & \quad \text{-----} \quad g
\end{align*}
\]

Contradiction!
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
&b^* \quad \quad \quad \quad g^* \\
&b \quad \quad \quad \quad g
\end{align*}
\]

\(b^*\) likes \(g^*\) more than \(g\).

By improvement lemma, \(g^*\) likes \(b^*\) better than \(b\).

Contradiction!
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \( (b, g^*) \)

\[
\begin{align*}
& b^* \quad g^* \\
& b \quad g
\end{align*}
\]

\( b \) likes \( g^* \) more than \( g \).

\( g^* \) likes \( b \) more than \( b^* \).
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
  b^* & \quad \text{likes} \quad g^* \quad \text{more than} \quad g. \\
  \quad \text{likes} \quad b \quad \text{more than} \quad b^*. \\
\end{align*}
\]

Boy \(b\) proposes to \(g^*\) before proposing to \(g\).
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
    b^* & \longrightarrow g^* \quad b \text{ likes } g^* \text{ more than } g. \\
    b & \longrightarrow g \quad g^* \text{ likes } b \text{ more than } b^*.
\end{align*}
\]

Boy \(b\) proposes to \(g^*\) before proposing to \(g\).
So \(g^*\) rejected \(b\) (since he moved on)
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
    b^* & \quad \longrightarrow \quad g^* & \quad b \text{ likes } g^* \text{ more than } g. \\
    b & \quad \longrightarrow \quad g & \quad g^* \text{ likes } b \text{ more than } b^*.
\end{align*}
\]

Boy \(b\) proposes to \(g^*\) before proposing to \(g\).
So \(g^*\) rejected \(b\) (since he moved on)
By improvement lemma, \(g^*\) likes \(b^*\) better than \(b\).
Pairing is Stable.

Lemma: There is no rogue couple for the pairing formed by traditional marriage algorithm.

Proof:
Assume there is a rogue couple; $(b, g^*)$

\[
\begin{array}{c}
\text{Boy } b \text{ proposes to } g^* \text{ before proposing to } g. \\
\text{So } g^* \text{ rejected } b \text{ (since he moved on)} \\
\text{By improvement lemma, } g^* \text{ likes } b^* \text{ better than } b. \\
\text{Contradiction!}
\end{array}
\]
Pairing is Stable.

**Lemma:** There is no rogue couple for the pairing formed by traditional marriage algorithm.

**Proof:**
Assume there is a rogue couple; \((b, g^*)\)

\[
\begin{align*}
  b^* & \quad \quad g^* \quad \quad b \text{ likes } g^* \text{ more than } g. \\
  b & \quad \quad g \quad \quad g^* \text{ likes } b \text{ more than } b^*.
\end{align*}
\]

Boy \(b\) proposes to \(g^*\) before proposing to \(g\).
So \(g^*\) rejected \(b\) (since he moved on)
By improvement lemma, \(g^*\) likes \(b^*\) better than \(b\).

Contradiction!
Good for boys? girls?

Is the TMA better for boys?
Good for boys? girls?

Is the TMA better for boys? for girls?
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is *x*-optimal if *x*'s partner is *x*'s best partner in any *stable* pairing.
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is *x-optimal* if *x*’s partner is *x*’s best partner in any stable pairing.

**Definition:** A pairing is *x-pessimal* if *x*’s partner is *x*’s worst partner in any stable pairing.
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is $x$-optimal if $x$’s partner is $x$’s best partner in any *stable* pairing.

**Definition:** A pairing is $x$-pessimal if $x$’s partner is $x$’s worst partner in any *stable* pairing.

**Definition:** A pairing is boy optimal if it is $x$-optimal for all boys $x$. 

Claim: Optimal partner for a boy must be first in his preference list.

True? False!

Subtlety here: Best partner in any stable pairing. As well as you can be in a globally stable solution!

Question: Is there a boy or girl optimal pairing?

Is it possible: $b$-optimal pairing different from the $b'$-optimal pairing!

Yes? No?
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is **x-optimal** if x’s partner is x’s best partner in any stable pairing.

**Definition:** A pairing is **x-pessimal** if x’s partner is x’s worst partner in any stable pairing.

**Definition:** A pairing is **boy optimal** if it is x-optimal for all boys x.

..and so on for boy pessimal, girl optimal, girl pessimal.
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is **x-optimal** if x’s partner is x’s best partner in any **stable** pairing.

**Definition:** A pairing is **x-pessimal** if x’s partner is x’s worst partner in any **stable** pairing.

**Definition:** A pairing is **boy optimal** if it is x-optimal for all boys x.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is *x-optimal* if *x*’s partner is *x*’s best partner in any stable pairing.

**Definition:** A pairing is *x-pessimal* if *x*’s partner is *x*’s worst partner in any stable pairing.

**Definition:** A pairing is **boy optimal** if it is *x-optimal* for all boys *x*.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True?
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is **x-optimal** if x’s partner is x’s best partner in any stable pairing.

**Definition:** A pairing is **x-pessimal** if x’s partner is x’s worst partner in any stable pairing.

**Definition:** A pairing is **boy optimal** if it is x-optimal for all boys x.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False?
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is $x$-optimal if $x$’s partner is $x$’s best partner in any stable pairing.

**Definition:** A pairing is $x$-pessimal if $x$’s partner is $x$’s worst partner in any stable pairing.

**Definition:** A pairing is boy optimal if it is $x$-optimal for all boys $x$.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A *pairing is* \( x \)-**optimal** if \( x \)'s partner is \( x \)'s best partner in any *stable* pairing.

**Definition:** A *pairing is* \( x \)-**pessimal** if \( x \)'s partner is \( x \)'s worst partner in any *stable* pairing.

**Definition:** A *pairing is* **boy optimal** if it is \( x \)-optimal for all boys \( x \).

..and so on for boy pessimal, girl optimal, girl pessimal.

**Claim:** Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any *stable* pairing.
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is \( x \)-**optimal** if \( x \)'s partner is \( x \)'s best partner in any **stable** pairing.

**Definition:** A pairing is \( x \)-**pessimal** if \( x \)'s partner is \( x \)'s worst partner in any **stable** pairing.

**Definition:** A pairing is **boy optimal** if it is \( x \)-optimal for **all** boys \( x \).

..and so on for boy pessimal, girl optimal, girl pessimal.

**Claim:** Optimal partner for a boy must be first in his preference list.

   True? False? False!

Subtlety here: Best partner in any **stable** pairing.

   As well as you can be in a globally stable solution!
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is $x$-optimal if $x$’s partner is $x$’s best partner in any stable pairing.

**Definition:** A pairing is $x$-pessimal if $x$’s partner is $x$’s worst partner in any stable pairing.

**Definition:** A pairing is boy optimal if it is $x$-optimal for all boys $x$.

..and so on for boy pessimal, girl optimal, girl pessimal.

**Claim:** Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any stable pairing.

As well as you can be in a globally stable solution!

**Question:** Is there a boy or girl optimal pairing?
Good for boys? girls?

Is the TMA better for boys? for girls?

Definition: A pairing is $x$-optimal if $x$’s partner is $x$’s best partner in any stable pairing.

Definition: A pairing is $x$-pessimal if $x$’s partner is $x$’s worst partner in any stable pairing.

Definition: A pairing is boy optimal if it is $x$-optimal for all boys $x$.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any stable pairing.

As well as you can be in a globally stable solution!

Question: Is there a boy or girl optimal pairing?

Is it possible:
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is *x*-optimal if *x*’s partner is *x*’s best partner in any stable pairing.

**Definition:** A pairing is *x*-pessimal if *x*’s partner is *x*’s worst partner in any stable pairing.

**Definition:** A pairing is boy optimal if it is *x*-optimal for all boys *x*.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any stable pairing.

As well as you can be in a globally stable solution!

Question: Is there a boy or girl optimal pairing?

Is it possible:

*b*-optimal pairing different from the *b*’-optimal pairing!
Is the TMA better for boys? for girls?

**Definition:** A pairing is **x-optimal** if x’s partner is x’s best partner in any stable pairing.

**Definition:** A pairing is **x-pessimal** if x’s partner is x’s worst partner in any stable pairing.

**Definition:** A pairing is **boy optimal** if it is x-optimal for all boys x.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any stable pairing.

As well as you can be in a globally stable solution!

Question: Is there a boy or girl optimal pairing?

Is it possible:

- \(b\)-optimal pairing different from the \(b'\)-optimal pairing!

Yes?
Good for boys? girls?

Is the TMA better for boys? for girls?

Definition: A pairing is \textit{x-optimal} if \(x\)'s partner is \(x\)'s best partner in any \textit{stable} pairing.

Definition: A pairing is \textit{x-pessimal} if \(x\)'s partner is \(x\)'s worst partner in any \textit{stable} pairing.

Definition: A pairing is \textit{boy optimal} if it is \(x\)-optimal for \textit{all} boys \(x\).

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any \textit{stable} pairing.

As well as you can be in a globally stable solution!

Question: Is there a boy or girl optimal pairing?

Is it possible:

\(b\)-optimal pairing different from the \(b'\)-optimal pairing!

Yes? No?
Good for boys? girls?

Is the TMA better for boys? for girls?

**Definition:** A pairing is *x*-optimal if *x*’s partner is *x*’s best partner in any **stable** pairing.

**Definition:** A pairing is *x*-pessimal if *x*’s partner is *x*’s worst partner in any **stable** pairing.

**Definition:** A pairing is **boy optimal** if it is *x*-optimal for **all** boys *x*.

..and so on for boy pessimal, girl optimal, girl pessimal.

Claim: Optimal partner for a boy must be first in his preference list.

True? False? False!

Subtlety here: Best partner in any **stable** pairing.

As well as you can be in a globally stable solution!

**Question:** Is there a boy or girl optimal pairing?

Is it possible:

*b*-optimal pairing different from the *b*’-optimal pairing!

Yes? No?
Understanding Optimality: by example.

A: 1,2       1: A,B
B: 1,2       2: B,A

Consider pairing: (A, 1), (B, 2).

Stable? Yes.
Optimal for B? Notice: only one stable pairing. So this is the best B can do in a stable pairing. So optimal for B.
Also optimal for A, 1 and 2.
Also pessimal for A, B, 1 and 2.

Pairing S: (A, 1), (B, 2).

Stable? Yes.
Pairing T: (A, 2), (B, 1).

Also Stable.
Which is optimal for A? S
Which is optimal for B? S
Which is optimal for 1? T
Which is optimal for 2? T
Understanding Optimality: by example.

A: 1,2  1: A,B
B: 1,2  2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.
Optimal for B? Yes.
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.
Also optimal for A, 1 and 2.

Pairing S: (A,1), (B,2).
Stable? Yes.
Pairing T: (A,2), (B,1).
Also Stable.
Which is optimal for A? S
Which is optimal for B? S
Which is optimal for 1? T
Which is optimal for 2? T
Understanding Optimality: by example.

A: 1,2  
B: 1,2

1: A,B  
2: B,A

Consider pairing: (A,1), (B,2).

Stable?
Understanding Optimality: by example.

A: 1,2  1: A,B  
B: 1,2  2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.
Understanding Optimality: by example.

A: 1,2  1: A,B
B: 1,2  2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A, 1), (B, 2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
Understanding Optimality: by example.

A: 1,2  1: A,B
B: 1,2  2: B,A

Consider pairing: (A, 1), (B, 2).

Stable? Yes.

Optimal for B?
  Notice: only one stable pairing.
  So this is the best B can do in a stable pairing.
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.
Understanding Optimality: by example.

A:  1,2        1:  A,B
B:  1,2        2:  B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2.
Understanding Optimality: by example.

A: 1,2  
B: 1,2

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.
Understanding Optimality: by example.

A: 1,2  1: A,B
B: 1,2  2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  1: B,A
B: 2,1  2: A,B
Understanding Optimality: by example.

A: 1,2
B: 1,2

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2
B: 2,1

Pairing S: (A,1), (B,2).
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B, 1 and 2.

A: 1,2 1: B,A
B: 2,1 2: A,B

Pairing S: (A,1), (B,2). Stable?
Understanding Optimality: by example.

A:  1,2  1:  A,B
B:  1,2  2:  B,A

Consider pairing: (A,1),(B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A:  1,2  1:  B,A
B:  2,1  2:  A,B

Pairing S: (A,1),(B,2). Stable? Yes.
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A,1),(B,2).

Stable? Yes.

Optimal for B?
  Notice: only one stable pairing.
  So this is the best B can do in a stable pairing.
  So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2 1: B,A
B: 2,1 2: A,B

Pairing S: (A,1),(B,2). Stable? Yes.
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A,1),(B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2 1: B,A
B: 2,1 2: A,B

Pairing S: (A,1),(B,2). Stable? Yes.

Pairing T: (A,2),(B,1).
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2 1: B,A
B: 2,1 2: A,B

Pairing S: (A,1), (B,2). Stable? Yes.

Pairing T: (A,2), (B,1). Also Stable.
Understanding Optimality: by example.

A: 1,2
B: 1,2

Consider pairing: (A,1),(B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2
B: 2,1

Pairing S: (A,1),(B,2). Stable? Yes.

Pairing T: (A,2),(B,1). Also Stable.

Which is optimal for A?
Understanding Optimality: by example.

A: 1,2  
   1: A,B
B: 1,2  
   2: B,A

Consider pairing: (A,1),(B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best $B$ can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  
   1: B,A
B: 2,1  
   2: A,B

Pairing $S$: (A,1),(B,2). Stable? Yes.

Pairing $T$: (A,2),(B,1). Also Stable.

Which is optimal for A? $S$
Understanding Optimality: by example.

A: 1,2  
   1: A,B
B: 1,2  
   2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  
   1: B,A
B: 2,1  
   2: A,B

Pairing S: (A,1), (B,2). Stable? Yes.
Pairing T: (A,2), (B,1). Also Stable.

Which is optimal for A? S
Which is optimal for B?
Understanding Optimality: by example.

A: 1,2  1: A,B
B: 1,2  2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  1: B,A
B: 2,1  2: A,B

Pairing S: (A,1), (B,2). Stable? Yes.

Pairing T: (A,2), (B,1). Also Stable.

Which is optimal for A? S Which is optimal for B? S
Understanding Optimality: by example.

A: 1,2 1: A,B
B: 1,2 2: B,A

Consider pairing: (A, 1), (B, 2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A, B, 1 and 2.

A: 1,2 1: B,A
B: 2,1 2: A,B

Pairing S: (A, 1), (B, 2). Stable? Yes.

Pairing T: (A, 2), (B, 1). Also Stable.

Which is optimal for A? S Which is optimal for B? S Which is optimal for 1?
Understanding Optimality: by example.

A: 1,2  
B: 1,2

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  
B: 2,1

Pairing S: (A,1), (B,2). Stable? Yes.
Pairing T: (A,2), (B,1). Also Stable.

Which is optimal for A? S       Which is optimal for B? S
Which is optimal for 1? T
Understanding Optimality: by example.

A:  1,2
B:  1,2

1: A,B 2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A:  1,2
B:  2,1

1: B,A 2: A,B

Pairing S: (A,1), (B,2). Stable? Yes.

Pairing T: (A,2), (B,1). Also Stable.

Which is optimal for A? S  Which is optimal for B? S
Which is optimal for 1? T  Which is optimal for 2?
Understanding Optimality: by example.

A: 1,2  
1: A,B
B: 1,2  
2: B,A

Consider pairing: (A,1), (B,2).

Stable? Yes.

Optimal for B?
Notice: only one stable pairing.
So this is the best B can do in a stable pairing.
So optimal for B.

Also optimal for A, 1 and 2. Also pessimal for A,B,1 and 2.

A: 1,2  
1: B,A
B: 2,1  
2: A,B

Pairing S: (A,1), (B,2). Stable? Yes.

Pairing T: (A,2), (B,1). Also Stable.

Which is optimal for A? S  
Which is optimal for B? S
Which is optimal for 1? T  
Which is optimal for 2? T
TMA is optimal for one gender!

For boys?
TMA is optimal for one gender!
For boys? For girls?

Theorem:
TMA produces a boy-optimal pairing.

Proof:
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^\ast$ - knocks $b$ off of $g$'s string on day $t$ $\Rightarrow g$ prefers $b^\ast$ to $b$

By choice of $t$, $b^\ast$ likes $g$ at least as much as optimal girl $\Rightarrow b^\ast$ prefers $g$ to his partner $g^\ast$ in $S$.

Rogue couple for $S$.

So $S$ is not a stable pairing.

Contradiction.

Notes: $S$ - stable.
$(b^\ast, g^\ast)$ and $(b, g) \in S$.
But $(b^\ast, g)$ is rogue couple!

Used Well-Ordering principle...
Induction.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**

Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ knocks $b$ off of $g$'s string on day $t = \Rightarrow g$ prefers $b^*$ to $b$.

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl. $= \Rightarrow b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$. So $S$ is not a stable pairing.

Contradiction.

Notes:

$S$ - stable.

$(b^*, g^*)$ and $(b, g) \in S$.

But $(b^*, g)$ is rogue couple!

Used Well-Ordering principle... Induction.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Notes:
S - stable.
((b\#, g\#) and (b, g) ∈ S).
But (b\#, g\#) is rogue couple!

Used Well-Ordering principle...
Induction.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$. 

Notes:
$S$ - stable.
$(b^*, g)$ and $(b, g) \in S$.
But $(b^*, g)$ is rogue couple! 

Used Well-Ordering principle... Induction.
TMA is optimal for one gender!
For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
- Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.
- There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
**TMA is optimal for one gender!**

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
by his optimal girl $g$ who he is paired with
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
by his optimal girl $g$ who he is paired with
in stable pairing $S$. 

Notes:
$S$ - stable.
$(b^*, g^*)$ and $(b, g) \in S$.
But $(b^*, g^*)$ is rogue couple!

Used Well-Ordering principle...
Induction.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
   Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

   There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
   by his optimal girl $g$ who he is paired with
   in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$
**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.
There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
by his optimal girl $g$ who he is paired with
in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t \implies g$ prefers $b^*$ to $b$
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
- Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
 Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
 by his optimal girl $g$ who he is paired with
 in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$. 
**TMA is optimal for one gender!**

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t \implies g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$. 
TMA is optimal for one gender!
For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy \( b \) not paired with optimal girl, \( g \) in TMA pairing \( T \).

There is a stable pairing \( S \) where \( b \) and \( g \) are paired.

Let \( t \) be first day a boy \( b \) gets rejected
by his optimal girl \( g \) who he is paired with
in stable pairing \( S \).

\( b^* \) - knocks \( b \) off of \( g \)'s string on day \( t \) \( \implies \) \( g \) prefers \( b^* \) to \( b \)

By choice of \( t \), \( b^* \) likes \( g \) at least as much as optimal girl.

\( \implies \) \( b^* \) prefers \( g \) to his partner \( g^* \) in \( S \).

Rogue couple for \( S \).
So \( S \) is not a stable pairing.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy \( b \) not paired with optimal girl, \( g \) in TMA pairing \( T \).

There is a stable pairing \( S \) where \( b \) and \( g \) are paired.

Let \( t \) be first day a boy \( b \) gets rejected

by his optimal girl \( g \) who he is paired with

in stable pairing \( S \).

\( b^* \) - knocks \( b \) off of \( g \)'s string on day \( t \)  \( \implies \) \( g \) prefers \( b^* \) to \( b \)

By choice of \( t \), \( b^* \) likes \( g \) at least as much as optimal girl.

\( \implies \) \( b^* \) prefers \( g \) to his partner \( g^* \) in \( S \).

Rogue couple for \( S \).
So \( S \) is not a stable pairing. Contradiction.
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected
by his optimal girl $g$ who he is paired with
in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.
$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.
Theorem: TMA produces a boy-optimal pairing. 
Proof: 
Assume not: 
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$. 

There is a stable pairing $S$ where $b$ and $g$ are paired. 

Let $t$ be first day a boy $b$ gets rejected 
by his optimal girl $g$ who he is paired with 
in stable pairing $S$. 

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$ 

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl. 

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$. 

Rogue couple for $S$. 
So $S$ is not a stable pairing. Contradiction. 

Notes:
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**

Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.

So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable.
Theorem: TMA produces a boy-optimal pairing.

Proof:
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t \implies g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b^*, g^*) \text{and} (b, g) \in S$. 
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$. There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$'s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b^*, g^*) and (b, g) \in S$. But $(b^*, g)$
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies$ $g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b^*, g^*) and (b, g) \in S$. But $(b^*, g)$ is rogue couple!
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:
Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t$ $\implies g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b^*, g^*)$ and $(b, g) \in S$. But $(b^*, g)$ is rogue couple!

Used Well-Ordering principle...
TMA is optimal for one gender!

For boys? For girls?

**Theorem:** TMA produces a boy-optimal pairing.

**Proof:**
Assume not:

Some boy $b$ not paired with optimal girl, $g$ in TMA pairing $T$.

There is a stable pairing $S$ where $b$ and $g$ are paired.

Let $t$ be first day a boy $b$ gets rejected by his optimal girl $g$ who he is paired with in stable pairing $S$.

$b^*$ - knocks $b$ off of $g$’s string on day $t \implies g$ prefers $b^*$ to $b$

By choice of $t$, $b^*$ likes $g$ at least as much as optimal girl.

$\implies b^*$ prefers $g$ to his partner $g^*$ in $S$.

Rogue couple for $S$.
So $S$ is not a stable pairing. Contradiction.

Notes: $S$ - stable. $(b^*, g^*) \text{and}(b, g) \in S$. But $(b^*, g)$ is rogue couple!

Used Well-Ordering principle...Induction.
How about for girls?

Theorem: TMA produces girl-pessimal pairing.

- \( T \) – pairing produced by TMA.
- \( S \) – worse stable pairing for girl \( g \).

In \( T \), \((g, b)\) is pair.

In \( S \), \((g, b^\ast)\) is pair.

\( g \) likes \( b^\ast \) less than she likes \( b \).

\( T \) is boy optimal, so \( b \) likes \( g \) more than his partner in \( S \).

\((g, b)\) is Rogue couple for \( S \).

\( S \) is not stable.

Contradiction.

Notes: Not really induction.

Structural statement: Boy optimality \( \Rightarrow \) Girl pessimality.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

- $T$ – pairing produced by TMA.
- $S$ – worse stable pairing for girl $g$.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.

$S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

*T* – pairing produced by TMA.

*S* – worse stable pairing for girl *g*.

In *T*, \((g, b)\) is pair.

In *S*, \((g, b^*)\) is pair.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.

$S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$. 
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

- $T$ – pairing produced by TMA.
- $S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

g likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$. 

How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

- $T$ – pairing produced by TMA.
- $S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.

$(g, b)$ is Rogue couple for $S$.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.
$S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.
In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.
$(g, b)$ is Rogue couple for $S$

$S$ is not stable.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

\( T \) – pairing produced by TMA.

\( S \) – worse **stable pairing** for girl \( g \).

In \( T \), \((g, b)\) is pair.

In \( S \), \((g, b^*)\) is pair.

\( g \) likes \( b^* \) less than she likes \( b \).

\( T \) is boy optimal, so \( b \) likes \( g \) more than his partner in \( S \).

\((g, b)\) is Rogue couple for \( S \)

\( S \) is not stable.

Contradiction.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.

$S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.

$(g, b)$ is Rogue couple for $S$

$S$ is not stable.

**Contradiction.**
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

T – pairing produced by TMA.

S – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.

$(g, b)$ is Rogue couple for $S$.

$S$ is not stable.

**Contradiction.**

Notes:
**Theorem:** TMA produces girl-pessimal pairing.

- $T$ – pairing produced by TMA.
- $S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.

$(g, b)$ is Rogue couple for $S$

$S$ is not stable.

**Contradiction.**

**Notes:** Not really induction.
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

$T$ – pairing produced by TMA.

$S$ – worse stable pairing for girl $g$.

In $T$, $(g, b)$ is pair.

In $S$, $(g, b^*)$ is pair.

$g$ likes $b^*$ less than she likes $b$.

$T$ is boy optimal, so $b$ likes $g$ more than his partner in $S$.

$(g, b)$ is Rogue couple for $S$.

$S$ is not stable.

**Contradiction.**

Notes: Not really induction.

   Structural statement: Boy optimality
How about for girls?

**Theorem:** TMA produces girl-pessimal pairing.

*T* – pairing produced by TMA.

*S* – worse stable pairing for girl *g*.

In *T*, *(g, b)* is pair.

In *S*, *(g, b*) is pair.

*g* likes *b* less than she likes *b*.

*T* is boy optimal, so *b* likes *g* more than his partner in *S*.

*(g, b)* is Rogue couple for *S*

*S* is not stable.

**Contradiction.**

Notes: Not really induction.

Structural statement: Boy optimality $\implies$ Girl pessimality.
Quick Questions.

How does one make it better for girls?
Quick Questions.

How does one make it better for girls?

SMA - stable marriage algorithm. One side proposes.
Quick Questions.

How does one make it better for girls?

- SMA - stable marriage algorithm. One side proposes.
- TMA - boys propose.
Quick Questions.

How does one make it better for girls?

SMA - stable marriage algorithm. One side proposes.
TMA - boys propose.
Girls could propose.
Quick Questions.

How does one make it better for girls?

- SMA: stable marriage algorithm. One side proposes.
- TMA: boys propose.
- Girls could propose.  \(\implies\) optimal for girls.
Residency Matching..
Residency Matching..

The method was used to match residents to hospitals.
The method was used to match residents to hospitals. Hospital optimal....
Residency Matching..

The method was used to match residents to hospitals. Hospital optimal.... ..until 1990’s..
Residency Matching..

The method was used to match residents to hospitals.
Hospital optimal....
..until 1990’s...Resident optimal.
The method was used to match residents to hospitals.
Hospital optimal....
..until 1990’s...Resident optimal.
Another variation: couples.
Residency Matching..

The method was used to match residents to hospitals.
Hospital optimal....
..until 1990’s...Resident optimal.
Another variation: couples.
Don’t go!

Summary.
Don’t go!

Summary.