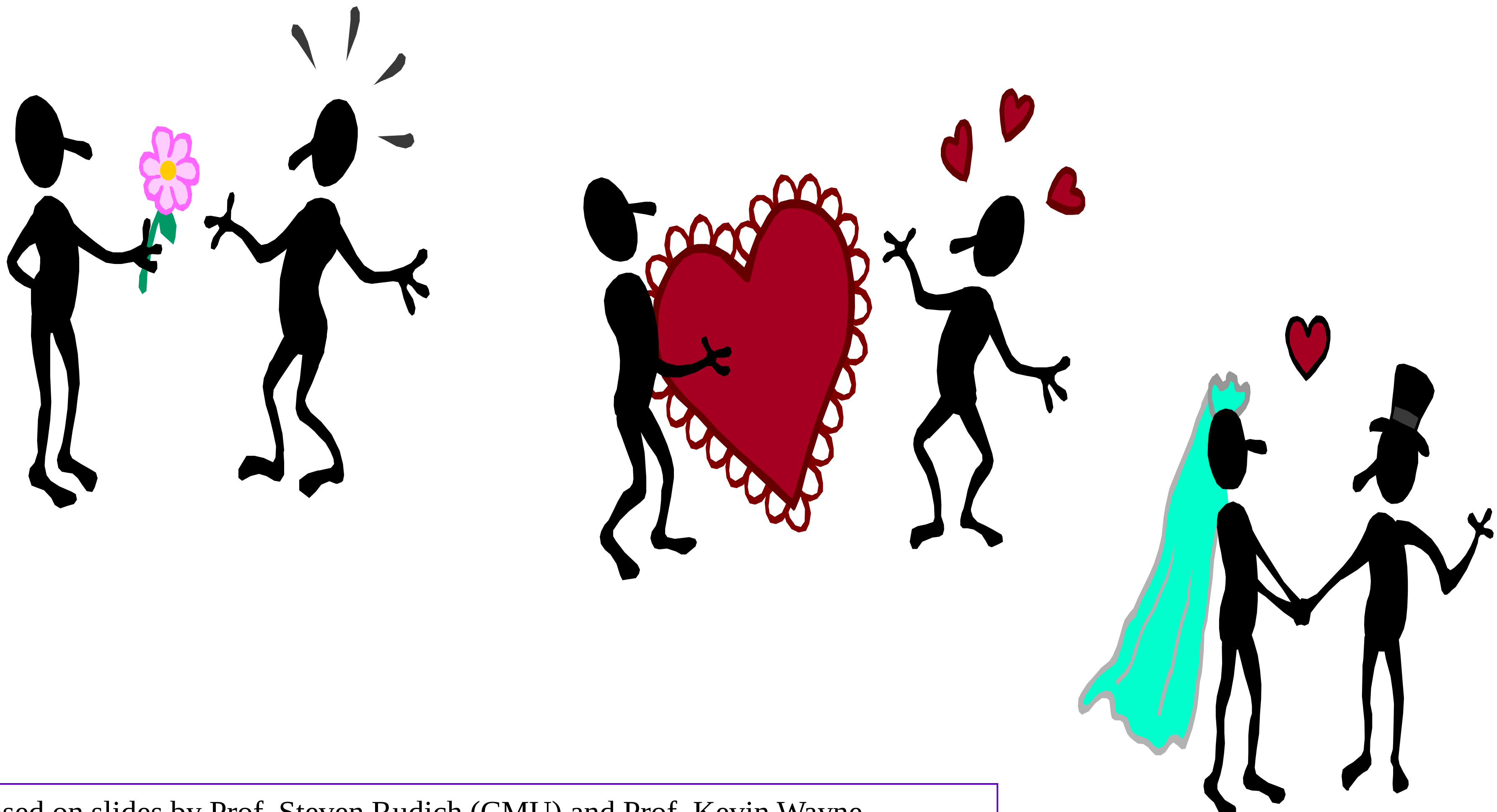
The Stable Matching Problem

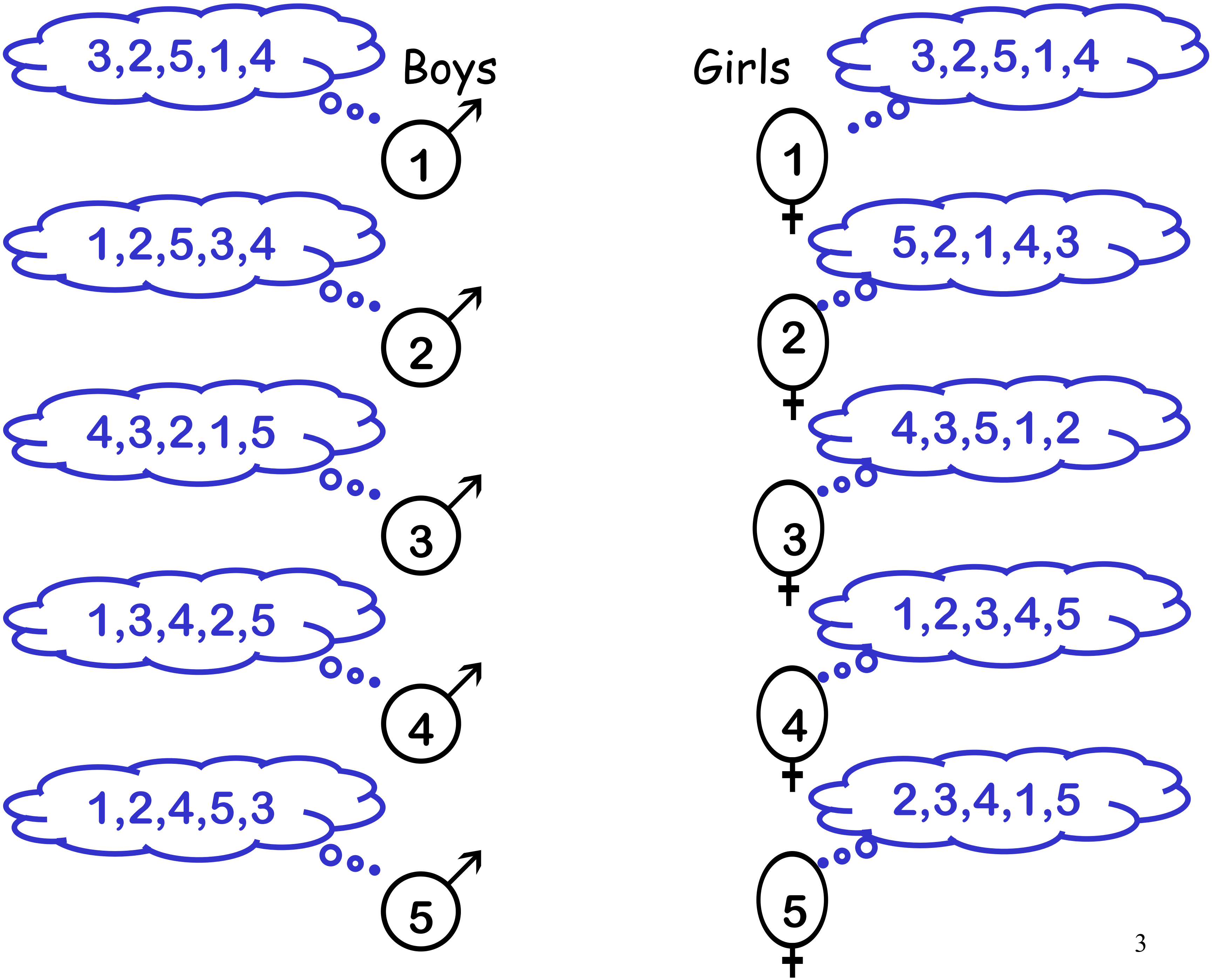


Dating Scenario

- There are n boys and n girls
- Each girl has her own ranked preference list of all the boys
- Each boy has his own ranked preference list of all the girls
- The lists have no ties

<u>Question:</u> How do we pair them off?

Which criteria come to mind?

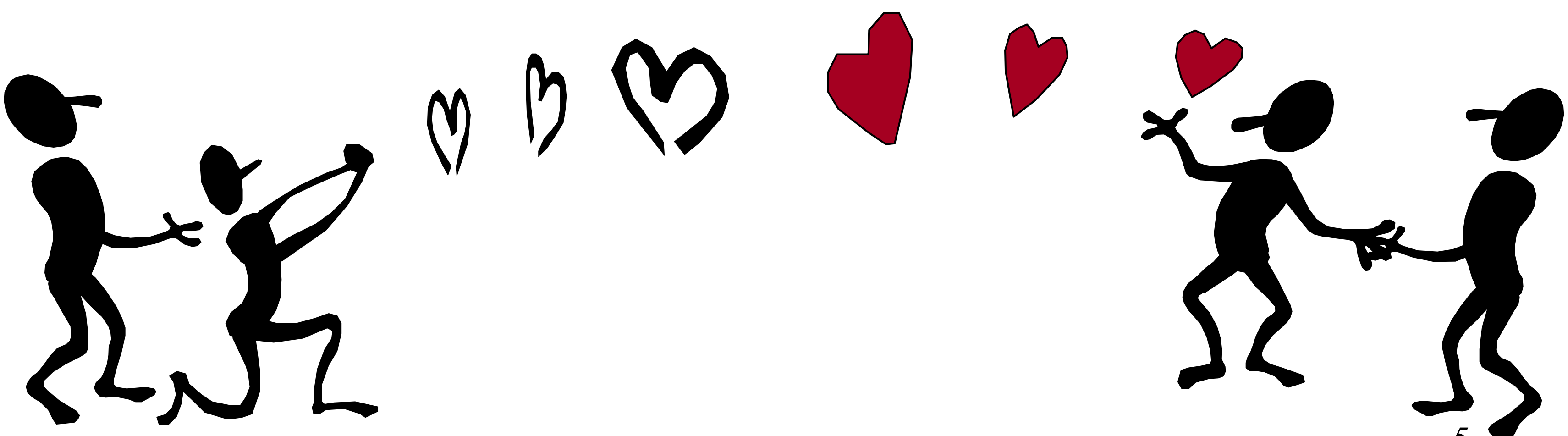

Boys
Girls
3,2,5,1,4
1
3,2,5,1,4
1
1,2,5,3,4
2
5,2,1,4,3
2
4,3,2,1,5
3
4,3,5,1,2
3
1,3,4,2,5
4
1,2,3,4,5
4
1,2,4,5,3
5
2,3,4,1,5
5

What is considered a “good” pairing?

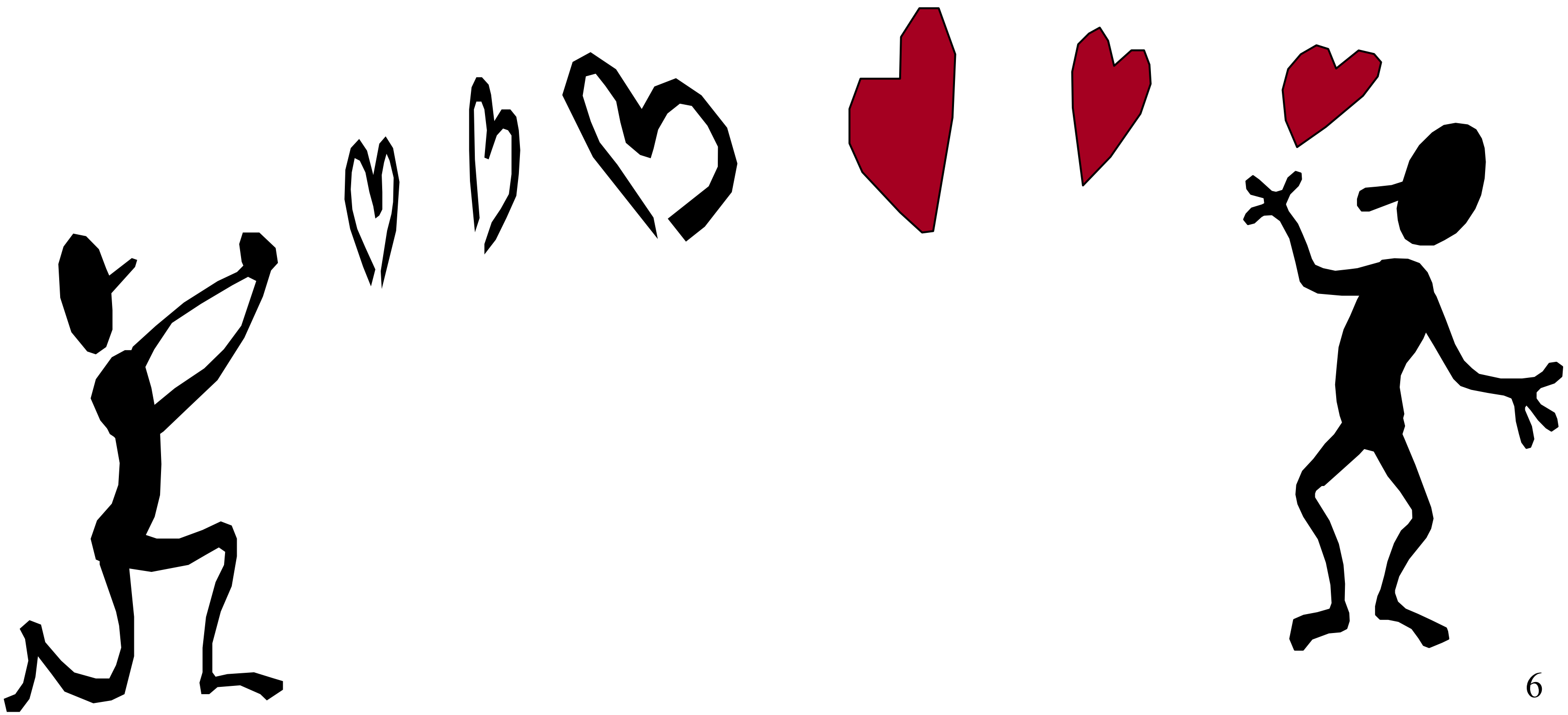
- Maximizing total satisfaction
  - What is the average rank of a match in the couple’s ranking.
- Maximizing the minimum satisfaction
  - How deep in their lists is the couple of the most unsatisfied participant?
- Minimizing the maximum difference in mate ranks
  - Everybody is more or less equally satisfied
- Maximizing the number of people who get their first choice

Rogue Couples

- Suppose we pair off all the boys and girls. Now suppose that some boy and some girl prefer each other to the people to whom they are paired. They will be called a <u>rogue couple</u>.



Why be with them when we can be with each other?



Stable Pairings

- A pairing of boys and girls is called <u>stable</u> if it contains no rogue couples.

Stability is a Must.

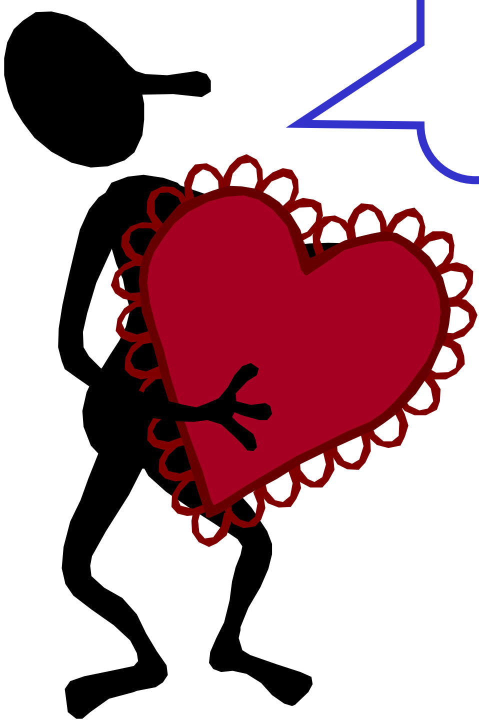
- Any list of criteria for a good pairing must include stability. (A pairing is doomed if it contains a rogue couple.)

- Any reasonable list of criteria must contain the stability criterion.

The study of stability will be the subject of this lecture.

- We will:
  - Analyze various mathematical properties of an algorithm that looks a lot like 1950's dating
  - Discover the naked mathematical truth about which sex has the romantic edge.
  - Go over additional application.
  - Taste some 'game theory'.

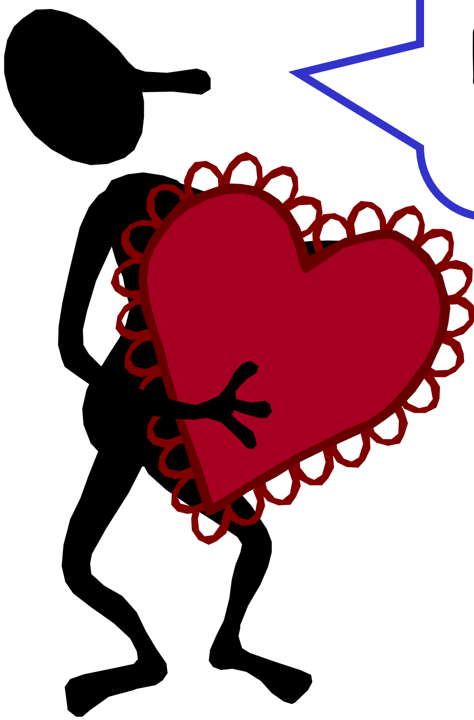
Given a set of preference lists, how do we find a stable pairing?



Wait! There is a more primary question!

<u>The Existence Question:</u>

Does every set of preferences lists have at least one stable pairing???



Can you argue that the couples will not continue breaking up and reforming forever?

An Instructive Variant: Roommate Problem

Stable roommate problem.

2n people; each person ranks others from 1 to 2n-1.

Assign roommates in a stable pairing.

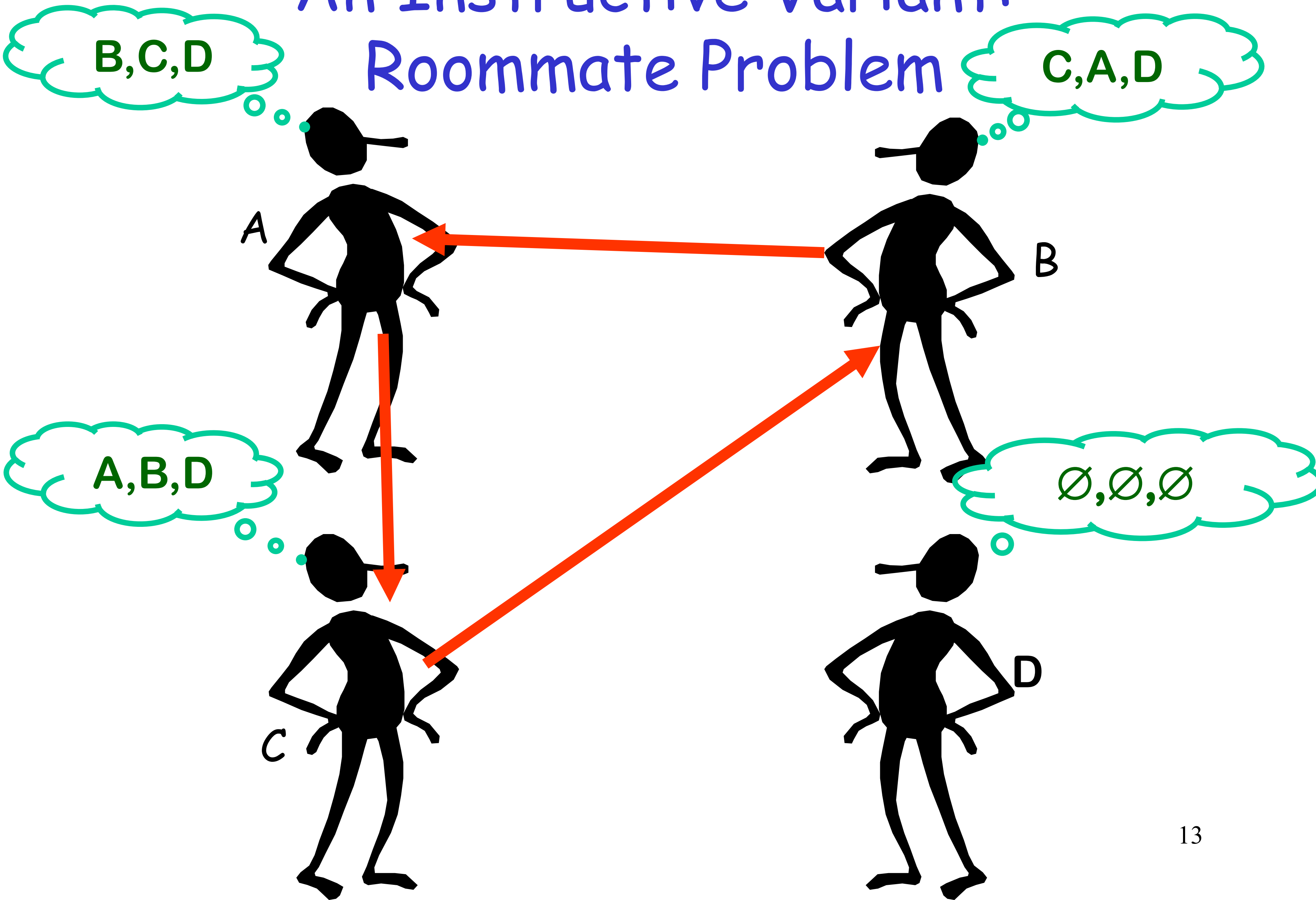
Observation: A stable matchings nay not exist.

	1st	2nd	3rd
Adam	B	C	D
Bob	C	A	D
Chris	A	B	D
Dan	A	B	C

A-B, C-D $\Rightarrow$ B-C unstable
A-C, B-D $\Rightarrow$ A-B unstable
A-D, B-C $\Rightarrow$ A-C unstable

Can be any order

An Instructive Variant: Roommate Problem

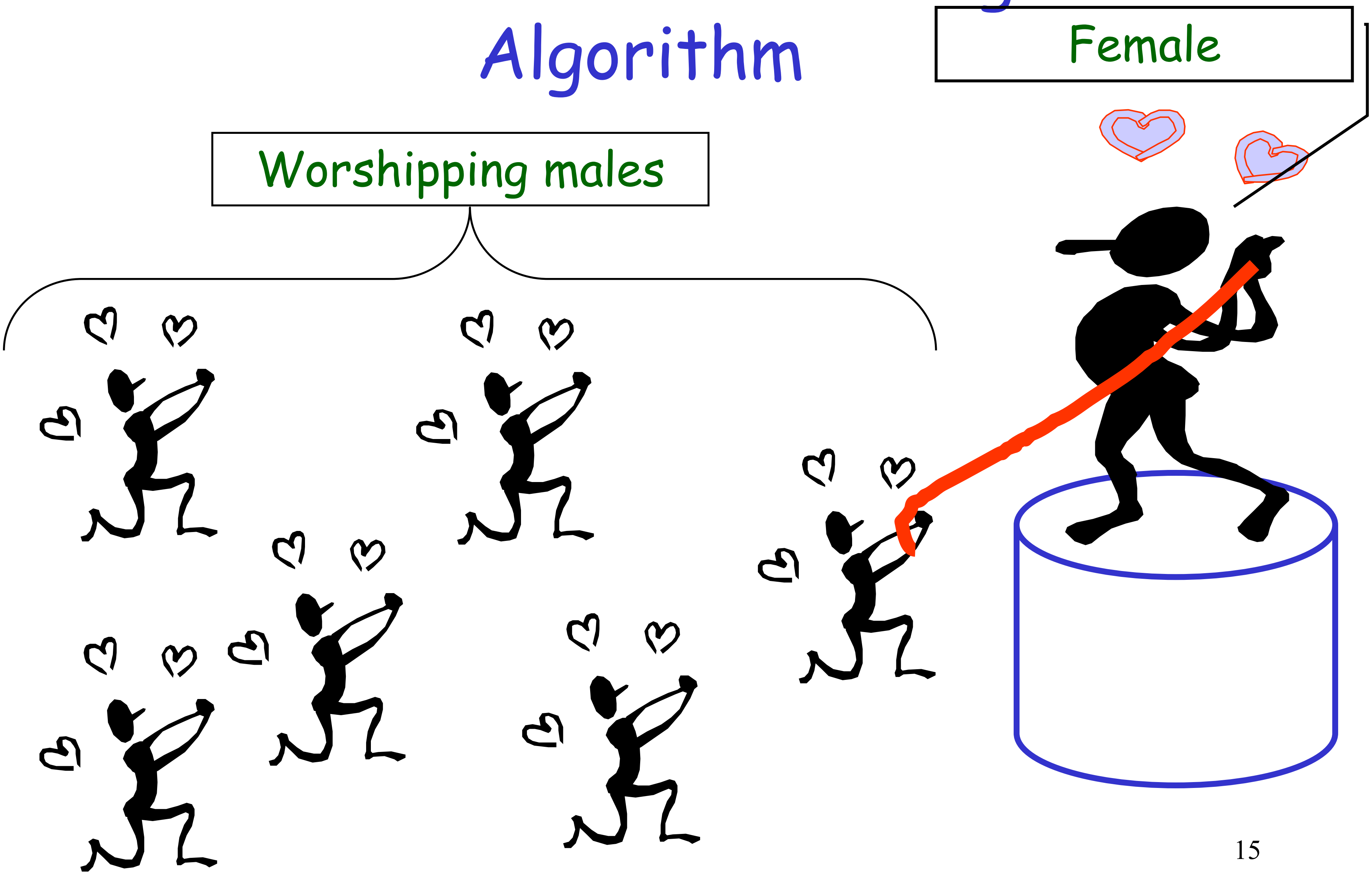


Insight

- Any proof that couples do not break up and reform forever must contain a step that fails in the case of the roommate problem.

- If you have a proof idea that works equally well in the marriage problem and the roommate problem, then your idea is not adequate to show the couples eventually stop.

The Traditional Marriage Algorithm



Traditional Marriage Algorithm

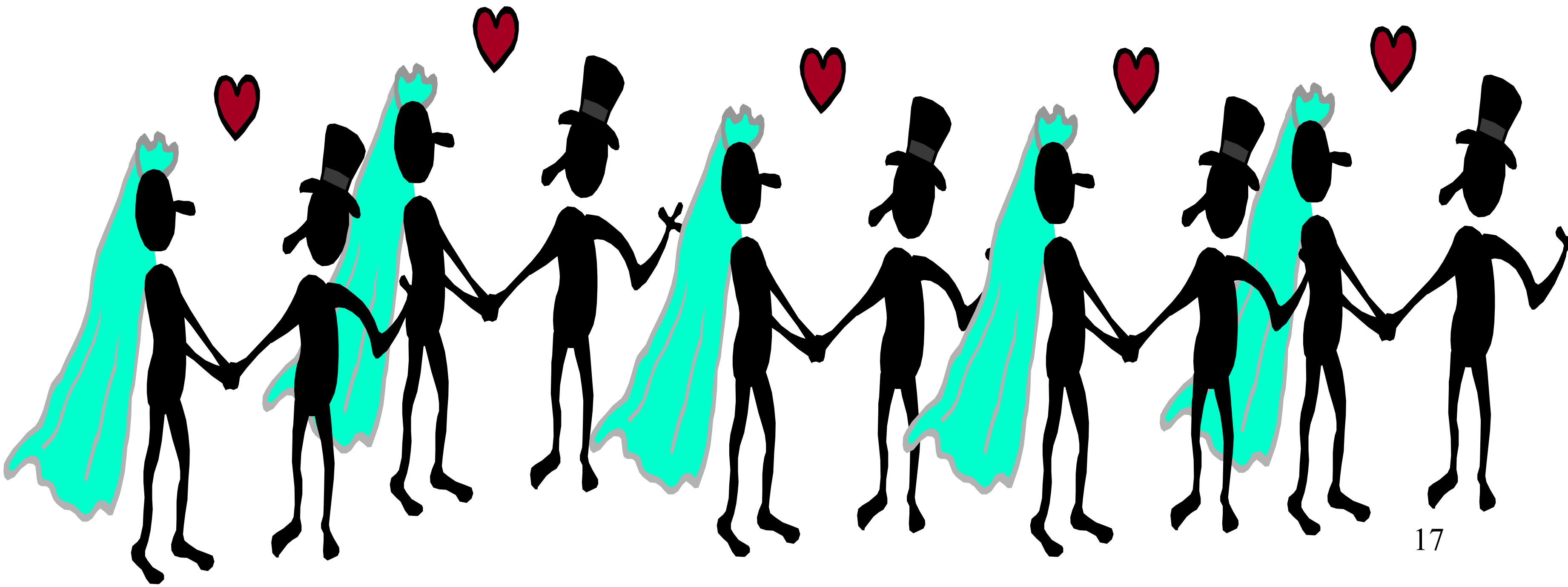
- For each day that some boy gets a "**No**" do:
  - **Morning**
    - Each girl stands on her balcony
    - Each boy proposes under the balcony of the best girl whom he has not yet crossed off
  - **Afternoon (for those girls with at least one suitor)**
    - To today's best suitor: **"Maybe, come back tomorrow"**
    - To any others: **"No, I will never marry you"**
  - **Evening**
    - Any rejected boy crosses the girl off his list.

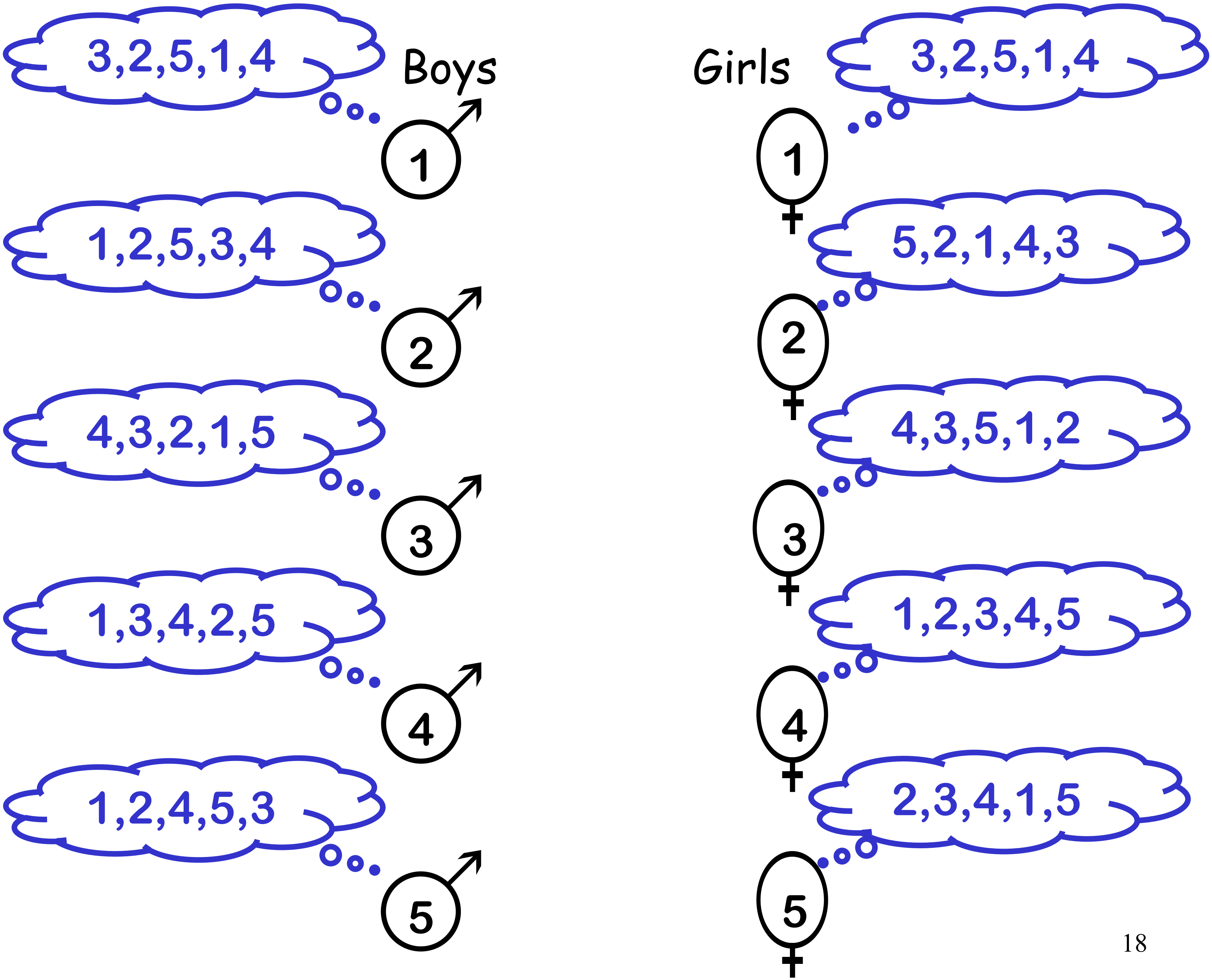
Each girl marries the boy to whom she just said "maybe"

Traditional Marriage Algorithm

Termination:

- When no boy get a **"No"** (all were told **"maybe"**):
- Each girl marries the boy to whom she said "maybe".





Boys
Girls
3,2,5,1,4
1
1,2,5,3,4
2
4,3,2,1,5
3
1,3,4,2,5
4
1,2,4,5,3
5
3,2,5,1,4
1
5,2,1,4,3
2
4,3,5,1,2
3
1,2,3,4,5
4
2,3,4,1,5
5

Traditional Marriage Algorithm

- Example (see prev. slide for preferences lists)

girl	Day 1	Day 2
1	2,4,5	2
2		5
3	1	1, 4
4	3	3
5		

○ = come tomorrow

In class exercise:

Complete the execution.

Traditional Marriage Algorithm

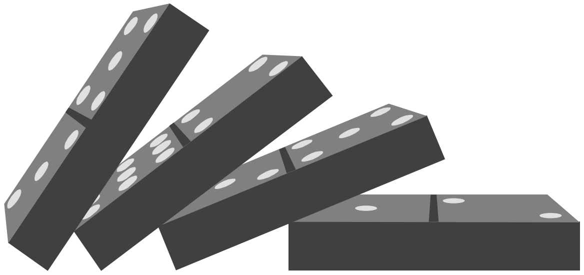
- Example (see prev. slide for preferences lists)

girl	Day 1	Day 2	Day 3	Day 4
1	2,4,5	2	2	2
2		5	5, 1	5
3	1	1, 4	4	4
4	3	3	3	3
5				1

Upgrade Lemma: In TMA, if on day i a girl says "maybe" to boy b, she is guaranteed to marry a husband that she likes at least as much as b.

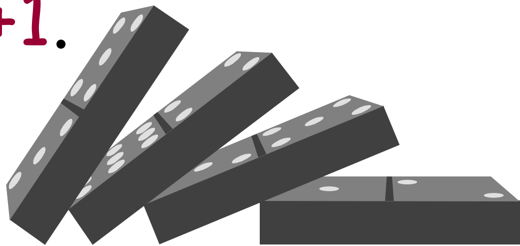
- She would only let him go in order to "maybe" someone better
- She would only let that guy go for someone even better
- She would only let that guy go for someone even better
- AND SO ON . . . .

Informal Induction



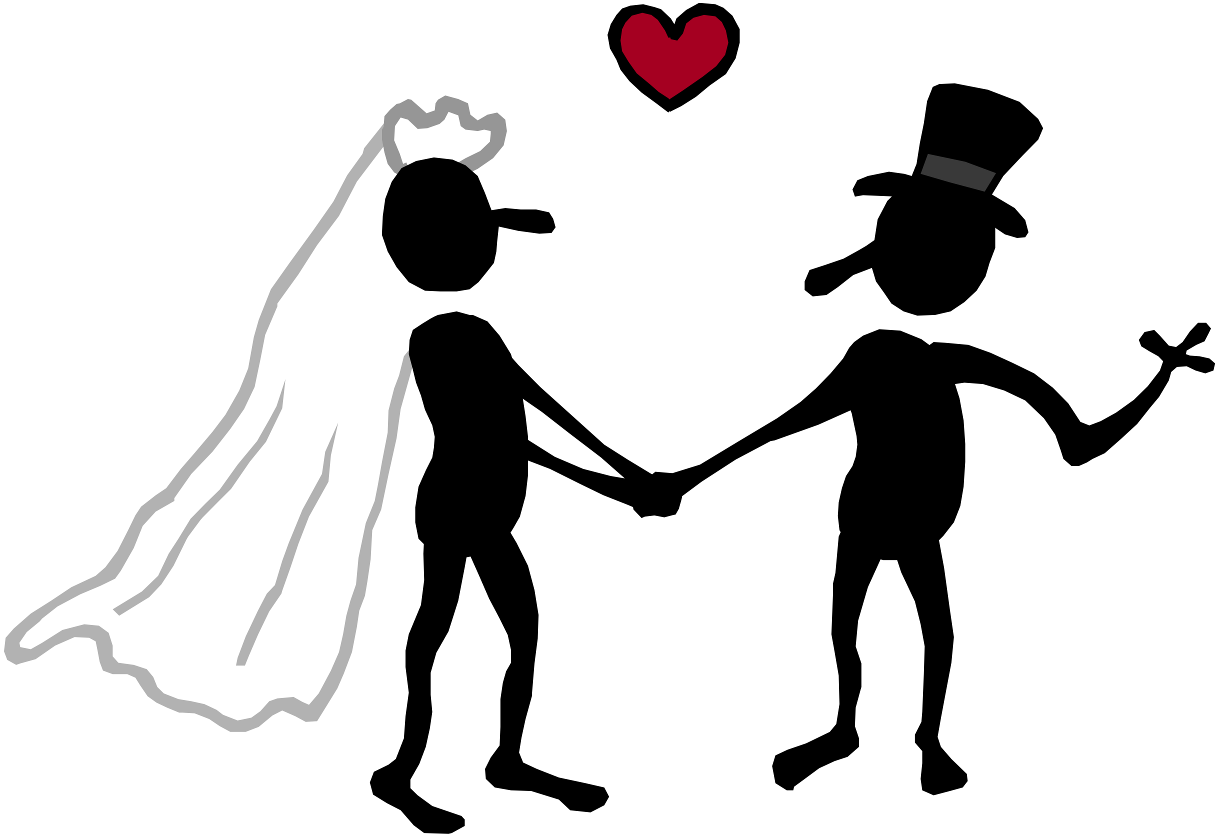
Upgrade Lemma: In TMA, if on day i a girl says "maybe" to boy b, she is guaranteed to marry a husband that she likes at least as much as b.

- (*) For all $k \geq 0$, on day $i+k$ the girl will say "maybe" to a boy she likes at least as much as b.
- Base: $k=0$ (true by assumption)
- Assume (*) is true for k. Thus she has a boy as good as b on day $i+k$. The next day she will either keep him or reject him for a better boy. Thus (*) is true for $k+1$.



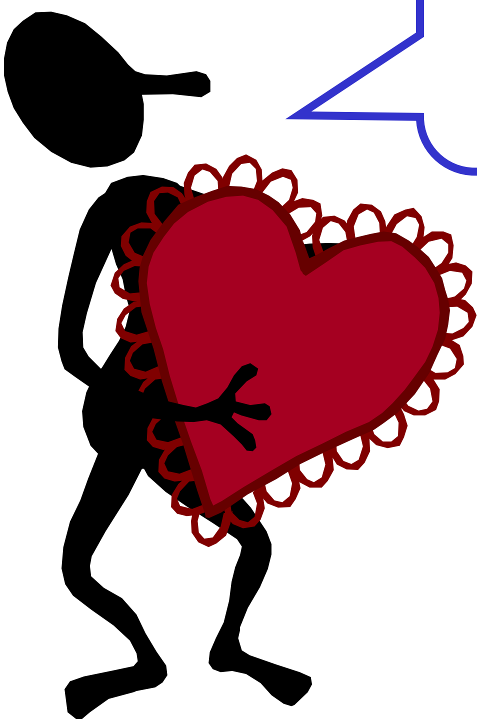
Formal Induction

Corollary: Each girl will marry her absolute favorite of the boys who visit her during the TMA.



Does the Traditional Marriage Algorithm always produce a stable pairing?

Wait! There is a more primary question!



Does TMA always terminate?

Does TMA always terminate?

- It might encounter a situation where the algorithm does not specify what to do next (core dump error).

- It might keep on going for an infinite number of days.

Lemma: Everyone will be matched.

- We show that no boy can be rejected by all the girls. This would imply that there are n couples.
- Suppose by contradiction that Bob is rejected by all the girls.
- Then some woman, say Amy, is not matched when Bob marked out the last girl from his list.
  - By the upgrade lemma, Amy was never proposed to.
  - But Bob proposes to everyone, since he ends up unmatched. In particular to Amy ▪

Contradiction

Theorem: The TMA always terminates in at most n^2 days

- Consider the "master list" containing all the boy's preference lists of girls. There are n boys, and each list has n girls on it, so there are a total of $n \times n = n^2$ girls' names in the master list.

- Each day that at least one boy gets a "No", at least one girl gets crossed off the master list.

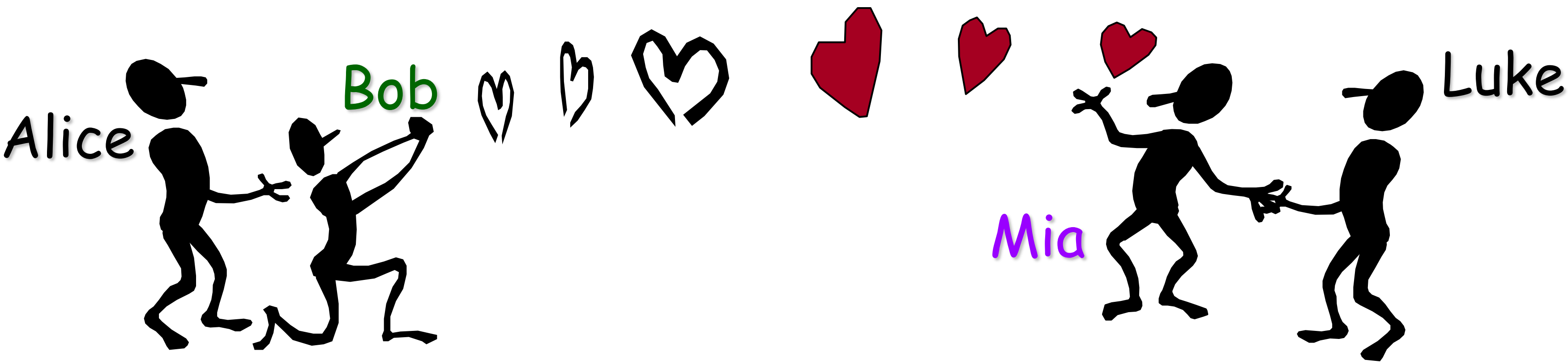
- Therefore, the number of days is bounded by the initial size of the master list, which in n^2.

Great! We know that TMA will terminate and produce a pairing.

But is it stable?

Theorem: The pairing produced by TMA is stable.

- Proof by contradiction:
  Suppose Bob and Mia are a rogue couple.



- This means Bob likes Mia more than his wife, Alice.
- Thus, Bob proposed to Mia before he proposed to Alice.
- Mia must have rejected Bob for someone she preferred.
- By the Upgrade lemma, she must like her husband Luke more than Bob.

Contradiction!

Opinion Poll

Understanding the Solution

- Question: For a given problem instance, there may be several stable matchings. Do all executions of the algorithm yield the same stable matching? If so, which one?

- An instance with two stable matchings:
  - A-X, B-Y, C-Z.
  - A-Y, B-X, C-Z.

	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

Forget TMA for a moment

- How should we define what we mean when we say "the optimal girl for Bob"?

Flawed Attempt:
"The girl at the top of Bob's list"

The Optimal Girl

- A boy's optimal girl is the highest ranked girl for whom there is some stable pairing in which the boy gets her.

- She is the best girl he can get in a stable world. Presumably, she may be better than the girl he gets in the stable pairing output by TMA.

The Pessimal Girl

- A boy's pessimal girl is the lowest ranked girl for whom there is some stable pairing in which the boy gets her.

- She is the worst girl he can get in a stable world.

Dating Heaven and Hell

- A pairing is male-optimal if every boy gets his optimal mate. This is the best of all possible stable worlds for all the boys simultaneously.

- A pairing is male-pessimal if every boy gets his pessimal mate. This is the worst of all possible stable worlds for all the boys simultaneously.

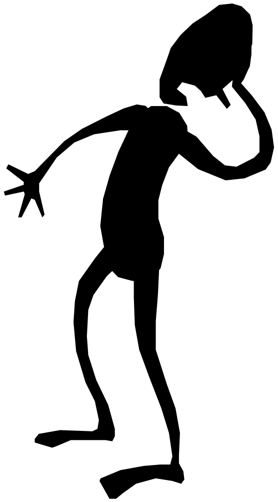
Dating Heaven and Hell

- A pairing is female-optimal if every girl gets her optimal mate. This is the best of all possible stable worlds for every girl simultaneously.

- A pairing is female-pessimal if every girl gets her pessimal mate. This is the worst of all possible stable worlds for every girl simultaneously.

The Naked Mathematical Truth!

- The Traditional Marriage Algorithm always produces a male-optimal, female-pessimal pairing.



Theorem: TMA produces a male-optimal pairing

- Suppose not: i.e. that some boy gets rejected by his optimal girl during TMA.
- In particular, let's say Bob is the first boy to be rejected by his optimal girl Mia: Let's say she said "maybe" to Luke, whom she prefers.
- Since Bob was the only boy to be rejected by his optimal girl so far, Luke must like Mia at least as much as his optimal girl.

The lists:

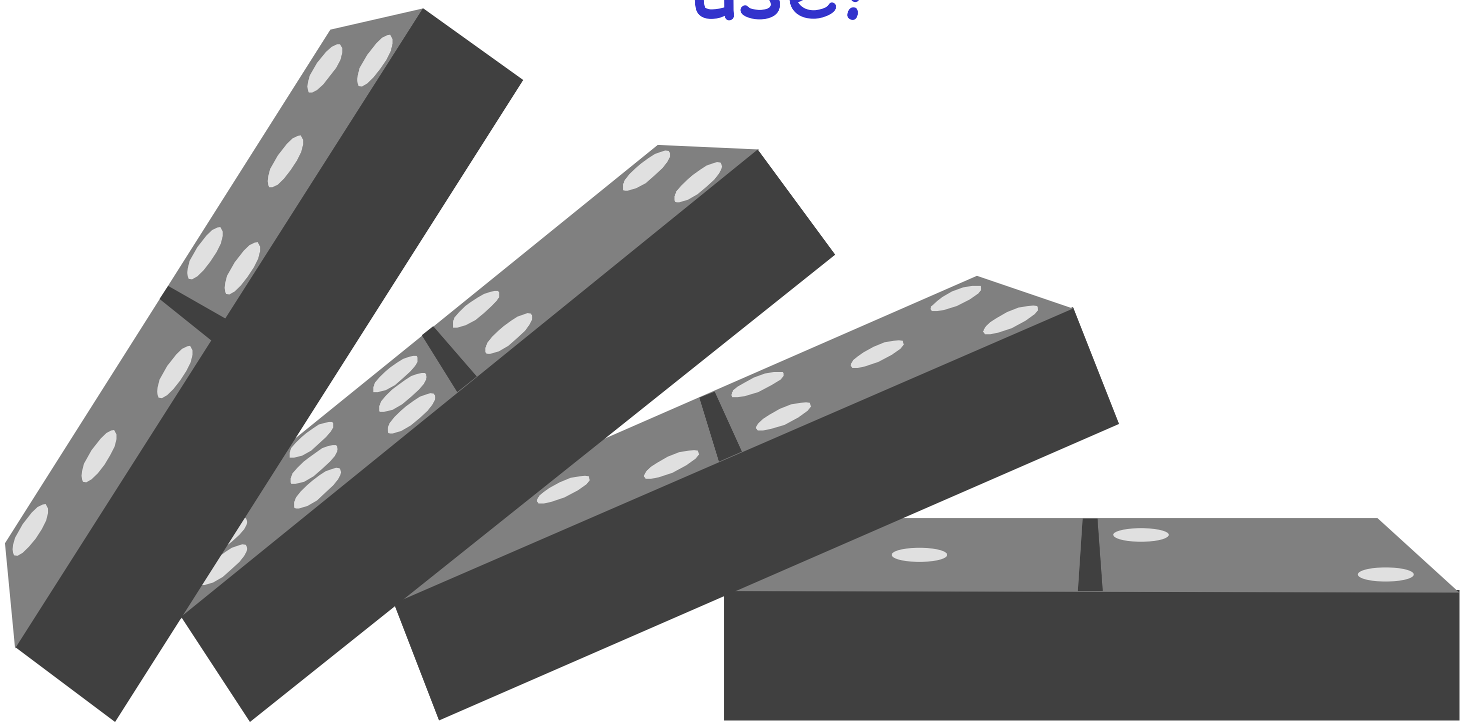
Mia: …. Luke ….. Bob…..

Luke: ….. Mia…. Luke's optimal girl….

Bob: …. Mia (Bob's optimal girl)…

- We show that any pairing S in which Bob marries Mia cannot be stable (for a contradiction).
- Suppose S is stable:
  - Luke likes Mia more than his wife in S (more than his optimal possible girl)
  - Mia likes Luke more than her husband Bob in S

What proof technique did we just use?



For every $t \geq 1$, no boy is rejected by his optimal girl during the first t days.

Theorem: The TMA pairing, T, is female-pessimal.

- Suppose there is a stable pairing S where some girl Alice does worse than in T.
- Let Luke be her mate in T.
  Let Bob be her mate in S.
  - Alice: … Luke….Bob…..
  - Luke likes Alice better than his mate in S.
    - We already know that Alice is his optimal girl (remember, T is male-optimal).
  - Therefore, S is not stable.

A contradiction

Efficient Implementation

- We describe an $O(n^2)$-time implementation.
- Men and women are denoted 1, …, n and 1', …, n' respectively.
- Engagements:
  - Maintain a list of free men, e.g., in a queue.
  - Maintain two arrays `wife[m]`, and `husband[w]`.
    - set entry to `0` if unmatched
    - if m is matched to w then `wife[m]=w` and `husband[w]=m`
- Men proposing:
  - For each man, maintain a list of women, ordered by preference.
  - Maintain an array `count[m]` that counts the number of proposals made by man `m`.

Efficient Implementation

- Women rejecting/accepting.
  - Does woman `w` prefer man `m` to man `m'`?
  - For each woman, create inverse of preference list of men.
  - Constant time access for each query after O(n) preprocessing.

Amy	1st	2nd	3rd	4th	5th	6th	7th	8th
prefBoy	8	3	7	1	4	5	6	2

Amy	1	2	3	4	5	6	7	8
inverse	4th	8th	2nd	5th	6th	7th	3rd	1st

Amy prefers man 3 to 6
since `inverse[3]` < `inverse[6]`
2 7

```
for i = 1 to n
   inverse[prefBoy[i]] = i
```

Efficient Implementation

- Initially, all men are unmatched.
- Iteration: Each unmatched man m proposes to `count[m]` (and `count[m]` increases)
- The women answer (each in $O(1)$)
- Arrays `husband/wife` updated. The queue is updated. $O(1)$ for each proposal made.
- Time complexity: Init: $O(n^2)$ (create `inverse[]`)
- n women, each woman is proposed at most n times.
- n men, man propose at most n times
- Total of at most $O(n^2)$ proposals, each proposal in $O(1)$.

Game Theory @ Stable Pairing Problem

- Question: Can there be an incentive to misrepresent your preference profile?
  - Assume you know TMA algorithm will be used.
  - Assume that you know the preference profiles of all other participants.

- Answer: No, for any man.
  Yes, for some women.

No mechanism can guarantee a stable matching and be cheatproof.

Game Theory @ Stable Pairing Problem

Example: Amy Lies and improves her match.

	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

Men's Preference List

	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

Women's True Preference Profile

	1st	2nd	3rd
Amy	Y	Z	X
Bertha	X	Y	Z
Clare	X	Y	Z

Amy Lies

References

- D. Gale and L. S. Shapley, *College admissions and the stability of marriage*, American Mathematical Monthly 69 (1962), 9-15
- Dan Gusfield and Robert W. Irving, *The Stable Marriage Problem: Structures and Algorithms*, MIT Press, 1989