15-251: Great Theoretical Ideas in Computer Science

Fall 2016 Lecture 11

October 4, 2016

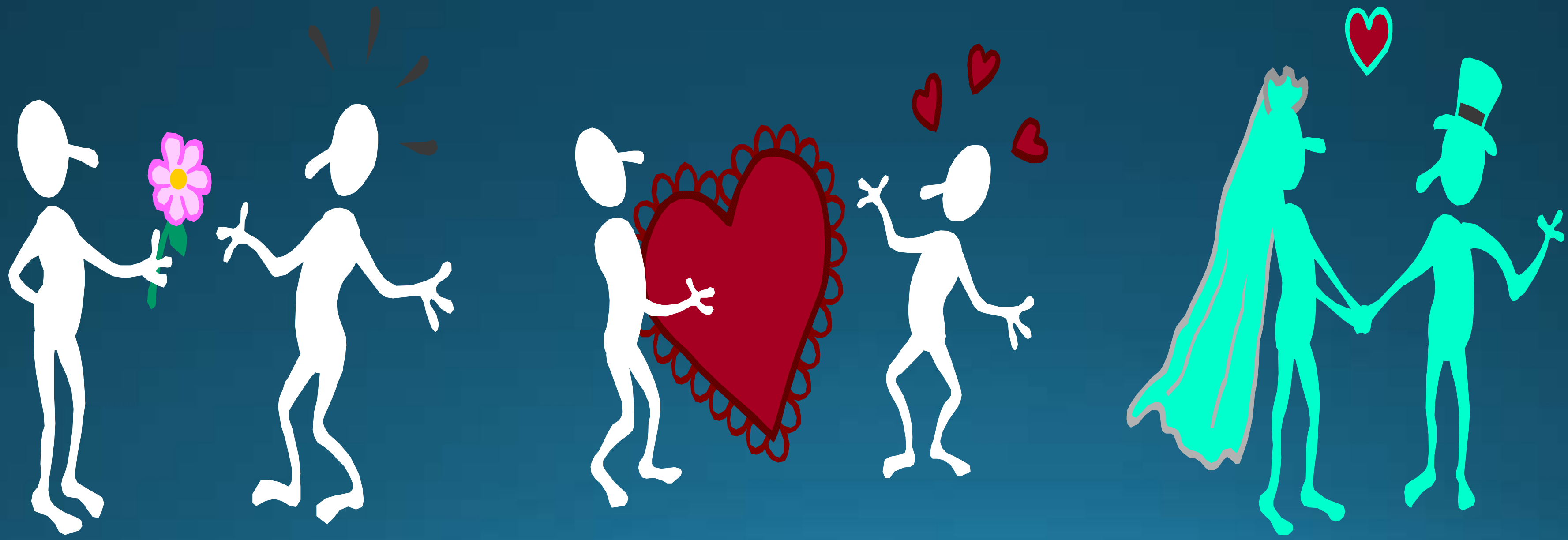
Graphs 3:
Stable Matchings

15-251: Great Theoretical Ideas in Computer Science

Fall 2016 Lecture 11

October 4, 2016

Dating and Marriage for Computer Scientists



First, a bit more about matchings

Matchings recap

A matching in a graph is a set of edges, no two of which share a vertex.

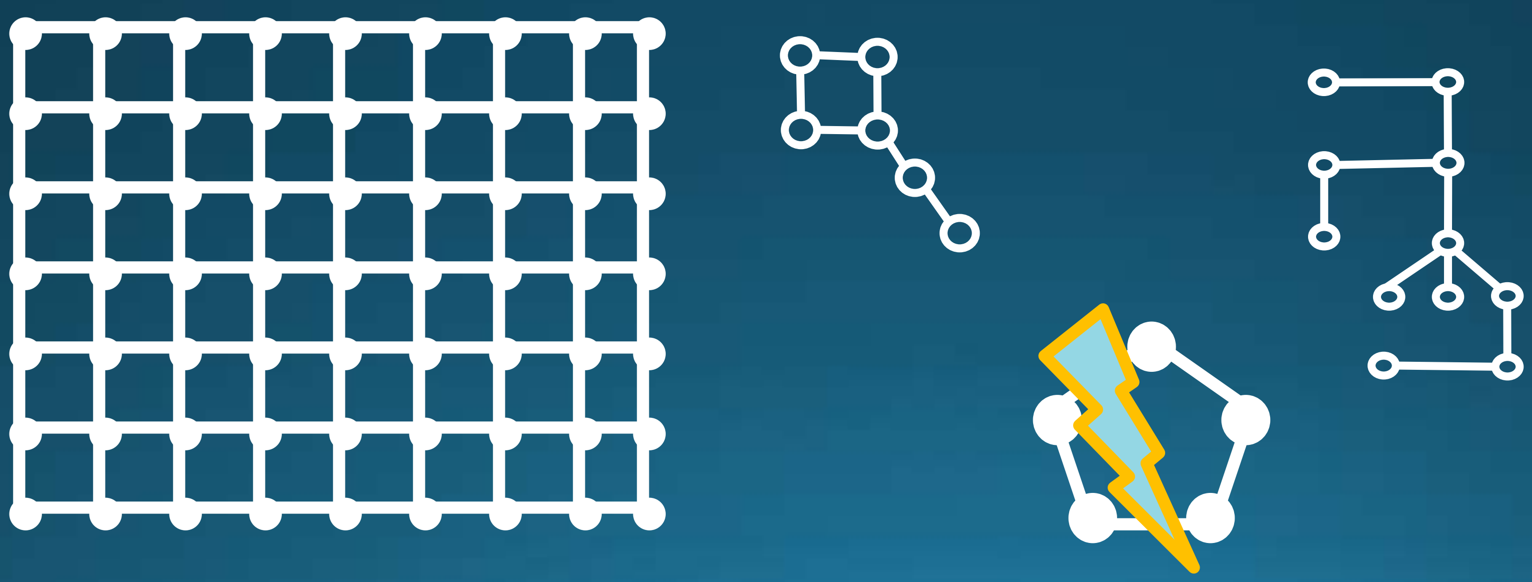
A matching is maximal if no edge can be added to it.

A matching is maximum if there is no matching with more edges.

A matching is perfect if it includes every vertex. (i.e., every vertex is matched)

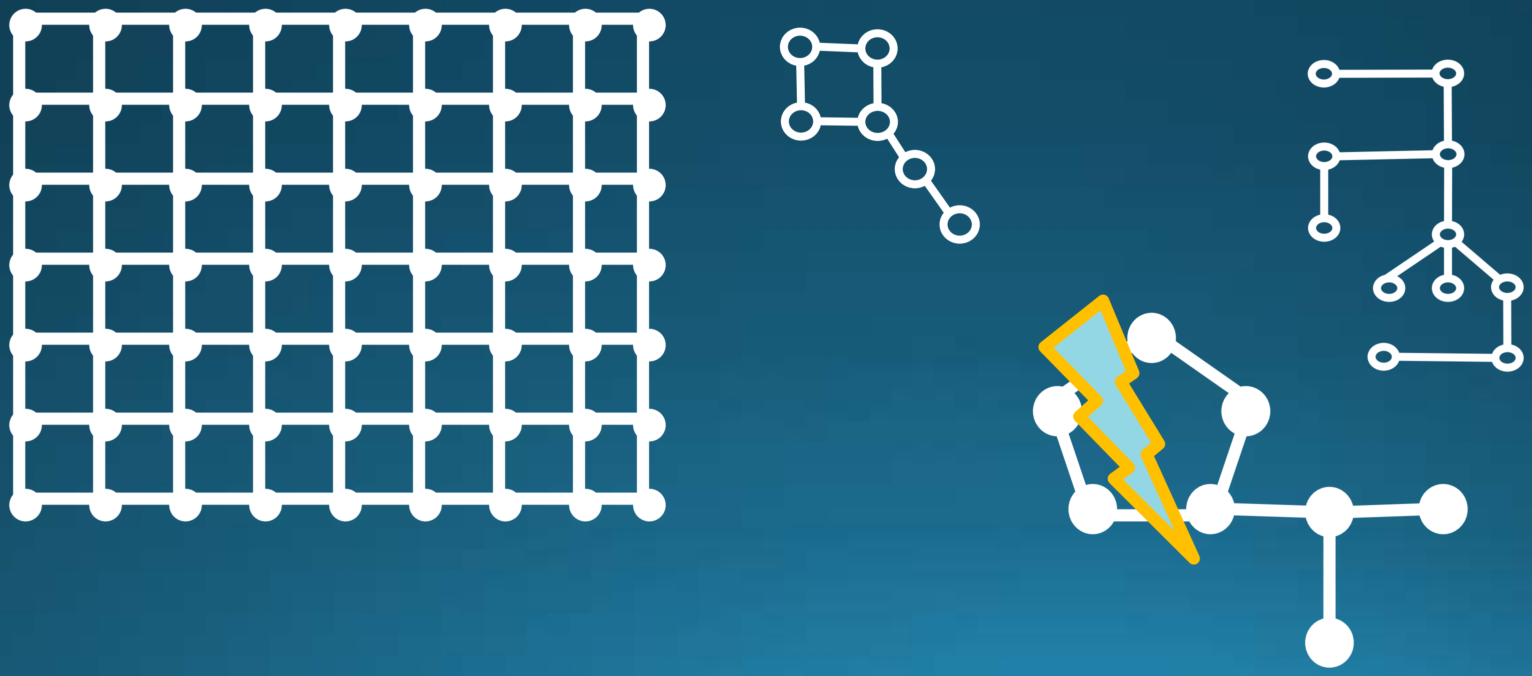
Bipartite Graphs

A graph is bipartite if it can be 2-colored, i.e., its nodes V can be partitioned into two sets R and B such that all edges go only between R and B.



Bipartite Graphs

A graph is bipartite if it can be 2-colored, i.e., its nodes V can be partitioned into two sets R and B such that all edges go only between R and B.



Bipartite Graph

Theorem: A graph is bipartite if and only if it contains no cycle of odd length.

Proof:

$\Longrightarrow$ The odd cycle cannot be 2-colored.

$\Longleftarrow$ Hint: Take BFS tree rooted at any node s.
The odd and even parts form the bipartition.

Note: This also gives an efficient $O(n+m)$ time algorithm to test bipartiteness and find a 2-coloring when the graph is bipartite.

Bipartite Maximum Matching

Theorem: There is a bipartite maximum matching algorithm running in $O(mn)$ time.

Algorithm:

- Start with any matching M (eg. empty, or any maximal one)
- Repeat finding augmenting paths till none exists (up to n iterations)
- Return the matching found as a maximum matching

Using Breadth First Search to Find an Alternating Path

The following algorithm starts with a matching M and either (1) determines that it is a maximum matching, or (2) constructs an alternating path to make the matching bigger.

Let L_0 be the set of unmatched vertices on the left.
For i=1,3,5 … do:

Let R_i be the vertices that have not yet been visited, which are neighbors of L_{i-1} via edges not in the matching.

If any vertex of R_i is not matched, we've found an alternating path. HALT

Let L_{i+1} be the vertices that have not yet been visited which are neighbors of R_i via edges in the matching.

If L_{i+1} is empty then our matching is maximum. HALT

Bipartite perfect matchings

Theorem: There is an algorithm computing a perfect matching in a bipartite graph in $O(mn)$ time, if one exists.

INTERESTING STRUCTURAL QUESTION:

For what graphs does such a perfect matching exist??

Hall's Marriage Theorem

Theorem:

A bipartite graph has a perfect matching

if and only if

$|A| = |B| = n$

and

For any $S \subseteq A$, there are at least $|S|$ nodes of B that are adjacent to a node in S.

Hall's Marriage Theorem

Theorem: A bipartite graph has a perfect matching if and only if $|A| = |B| = n$ and for any $S \subseteq A$, there are at least $|S|$ nodes of B that are adjacent to a node in S.

Proof: $\Longrightarrow$ (only if) obvious

$\Longleftarrow$ (if) (Hint):

- Let M be a *maximum* matching (but not a perfect matching)
- Look at the "modified BFS-tree" rooted at an unmatched node in A.
- As there is no augmenting path, infer that odd level nodes give a subset $S \subseteq A$ with fewer than $|S|$ neighbors in B.

Note: Can also prove using induction

Hall’s theorem is a very handy way to argue the existence of perfect matchings in graphs

Example

Suppose that a standard deck of cards is dealt in an arbitrary manner into 13 piles of 4 cards each

Then it is always possible to select a card from each pile so that the 13 chosen cards contain exactly one card of each rank

Proof: Form a bipartite graph as follows: Start with 52 cards on the left and the same 52 cards on the right, connected by 52 edges.

Now group the cards on the left into 13 sets according to the given piles. Group the cards on the right into 13 groups according to rank. Let the edges be inherited from the original ones (there can be multiple edges between nodes)

This bipartite graph satisfies condition of Hall's theorem -- k groups on the left have to connect to 4k cards on the right, thus they connect to at least k groups on the right.

And thus it has a perfect matching.

Today’s main topic:
Stable matchings

Discretion is advised

WARNING: This lecture contains mathematical content that may be shocking

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 the girls

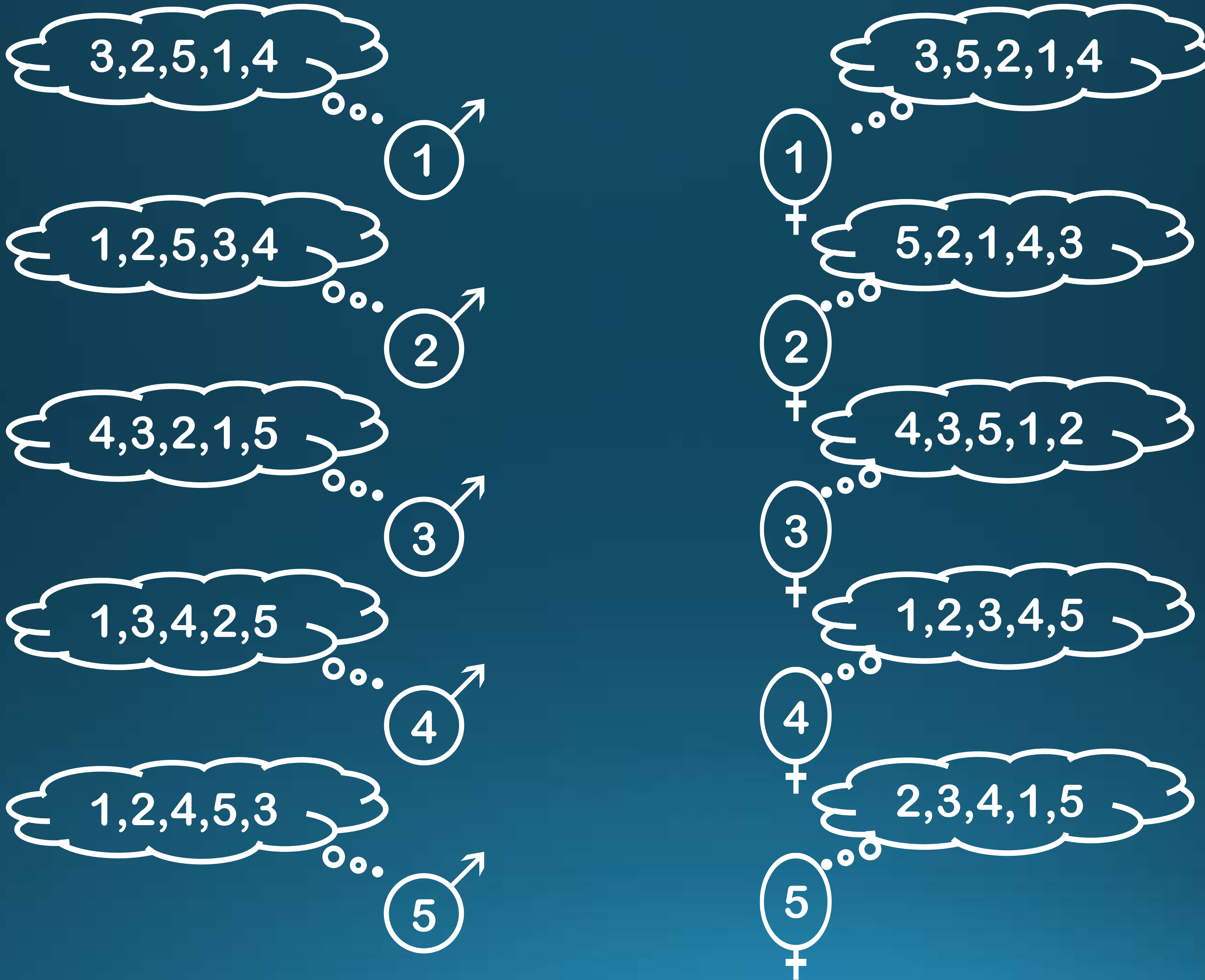
The lists have no ties

Question: How do we pair them?

Matchings - Context

While we'll use the dating metaphor, what we will discuss now is applicable and used in matching hospitals and residents (matchingMarkets package)

CSD handshake process (for matching first year Ph.D. students and advisors, coincidentally announced at today's "login ball") takes preference lists as input to the process.


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
3,5,2,1,4
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

More Than One Notion of What Constitutes A “Good” Pairing

Maximizing total satisfaction

Maximizing the minimum satisfaction

Minimizing maximum difference in mate ranks

Maximizing number of people getting their first choice

We will ignore the issue of what is “best”!

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 rogue couple

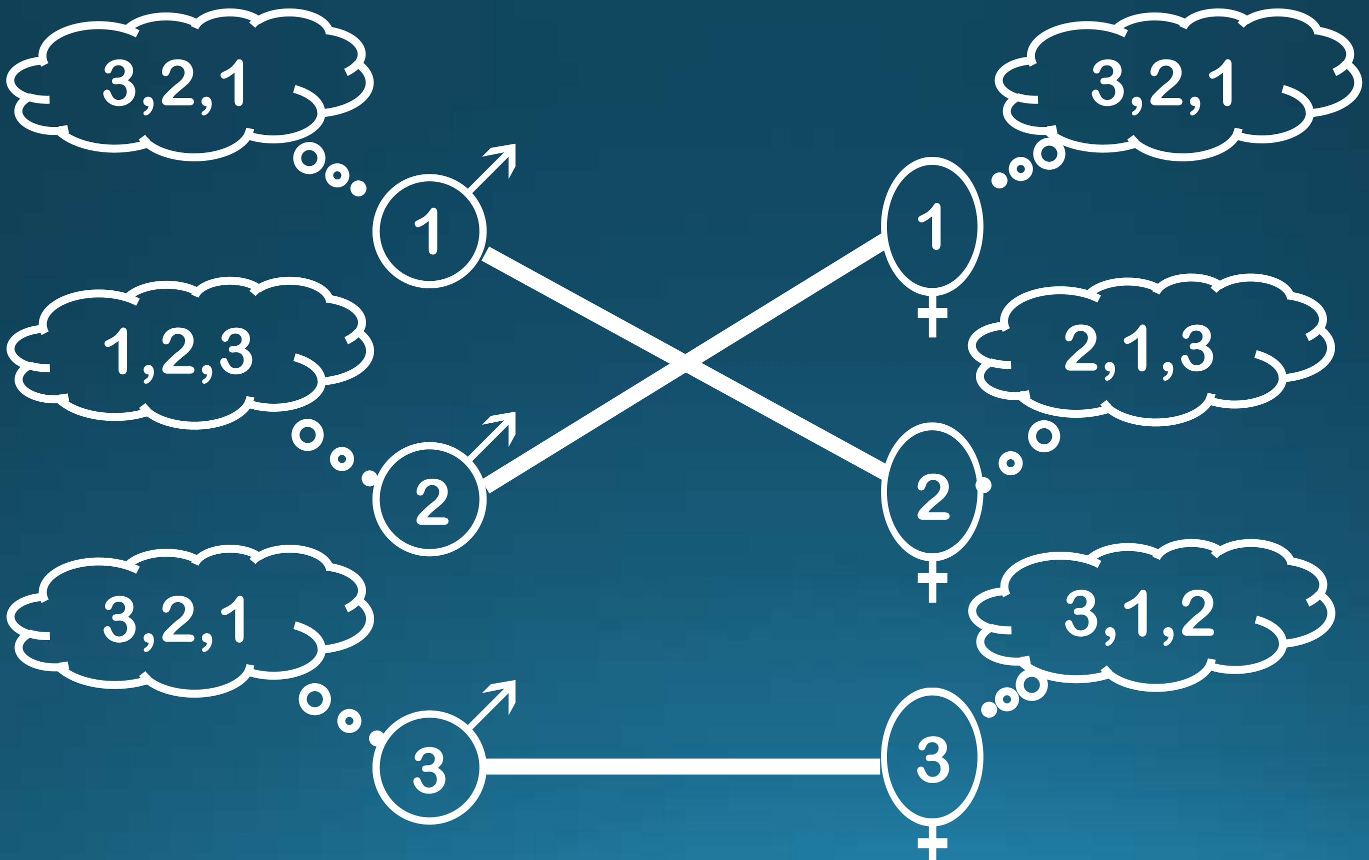
Why be with them when we can be with each other?

What use is fairness, if it is not stable?

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

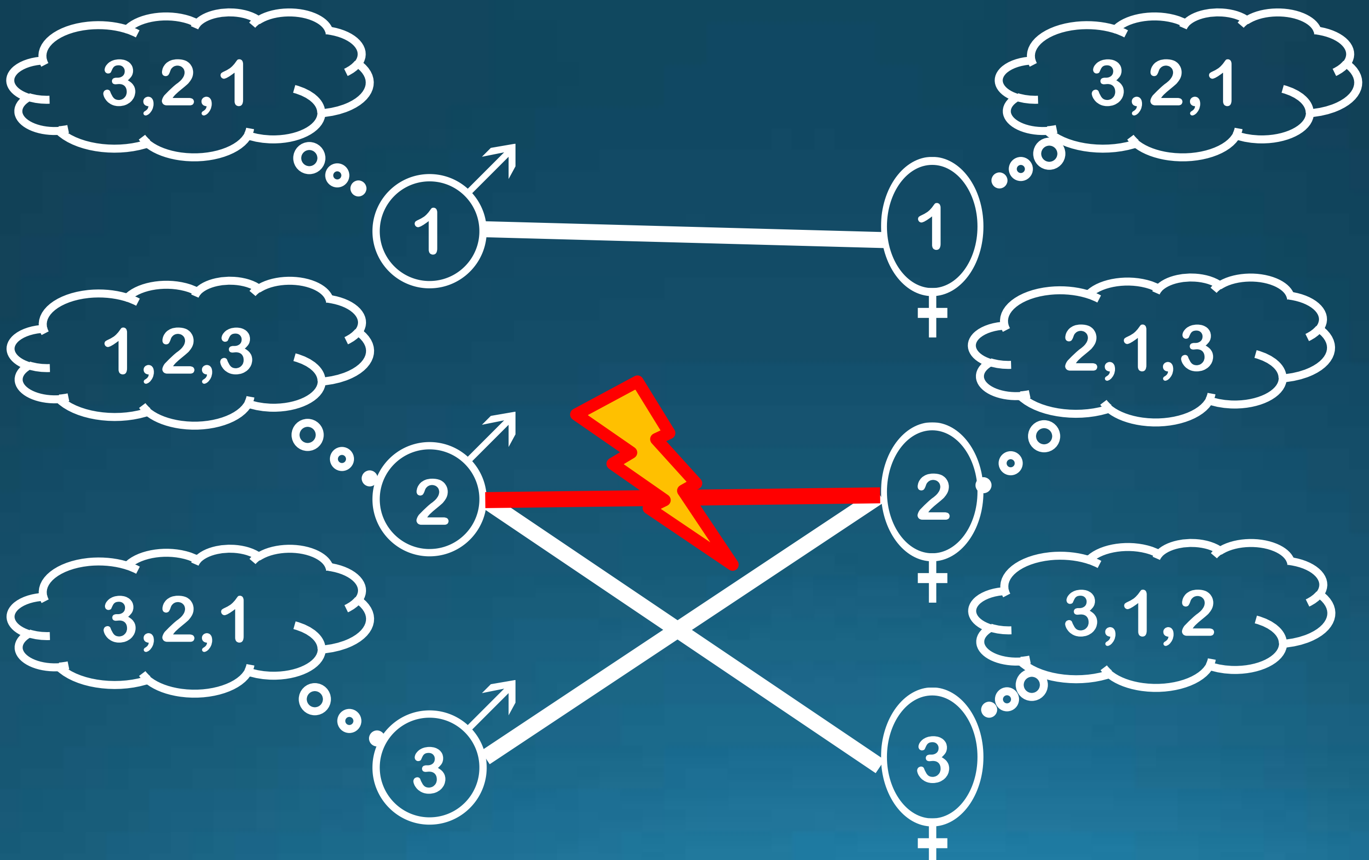
Stable Pairings

A pairing of boys and girls is called stable if it contains no rogue couples



Stable Pairings

A pairing of boys and girls is called unstable if it contains at least one rogue couple



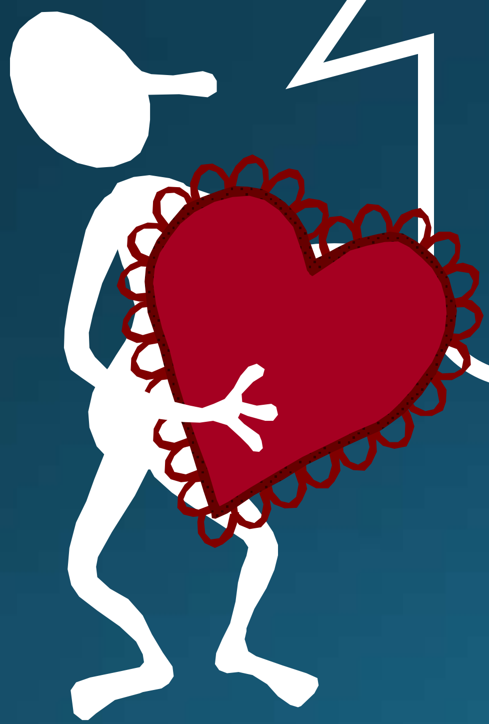
Stability is the subject of the remaining lecture

We will:

Analyze various mathematical properties of an algorithm that looks a lot like 1950's dating

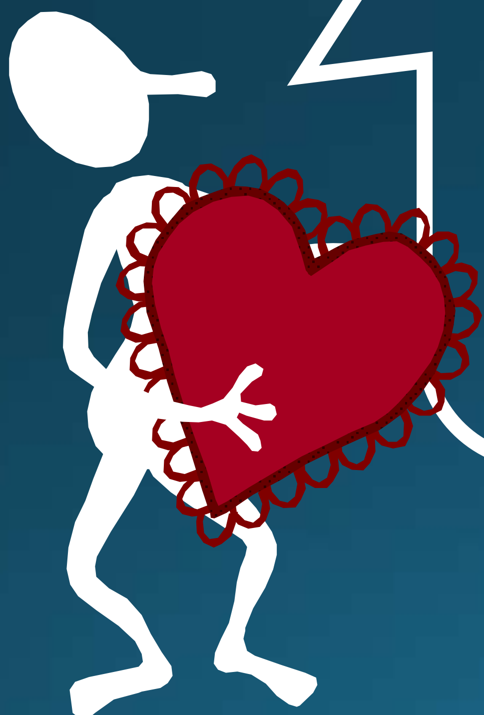
Discover the *mathematical truth* about which sex has the romantic edge

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



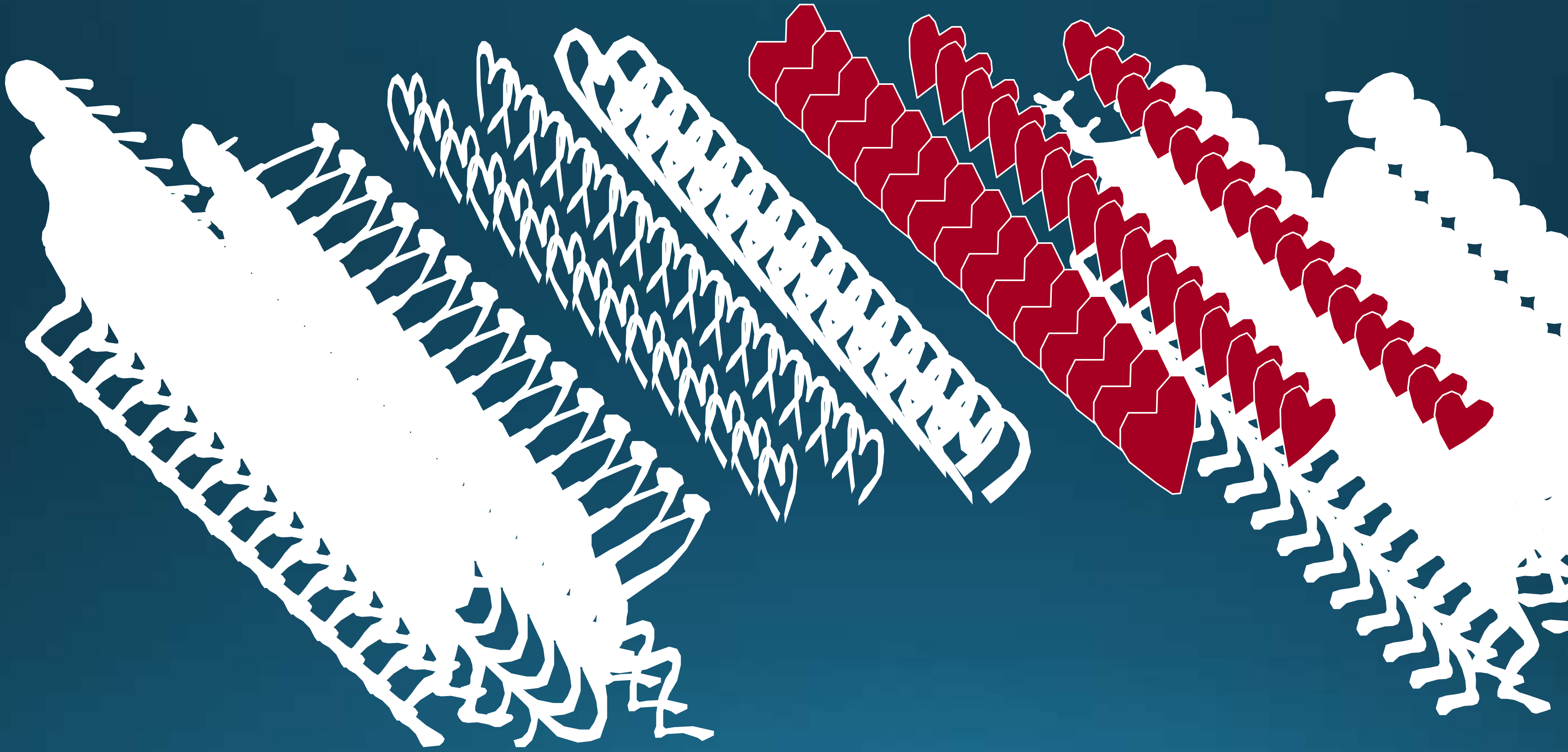
Wait! We don’t even know that such a pairing always exists!

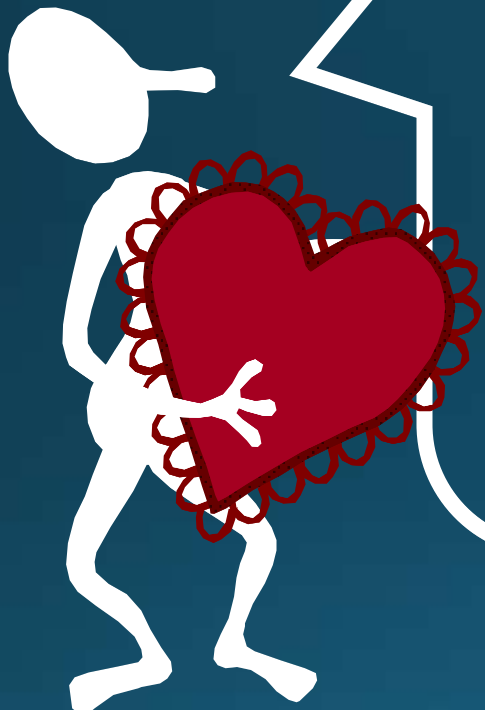
Revised Question



Does every set of preference lists have a stable pairing?

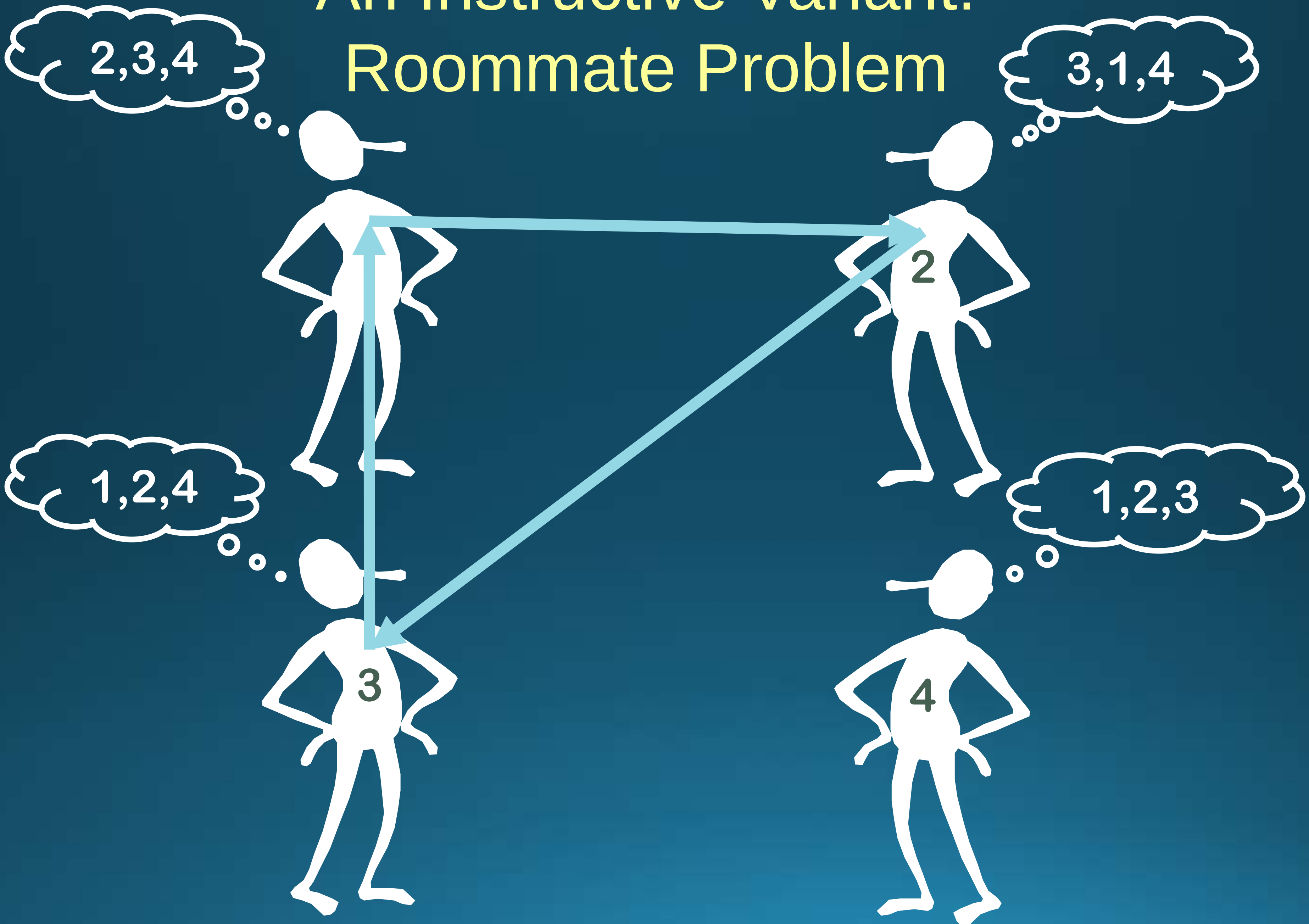
Idea: Allow the pairs to keep breaking up and reforming until they become stable





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

An Instructive Variant: Roommate Problem



Stable roommates

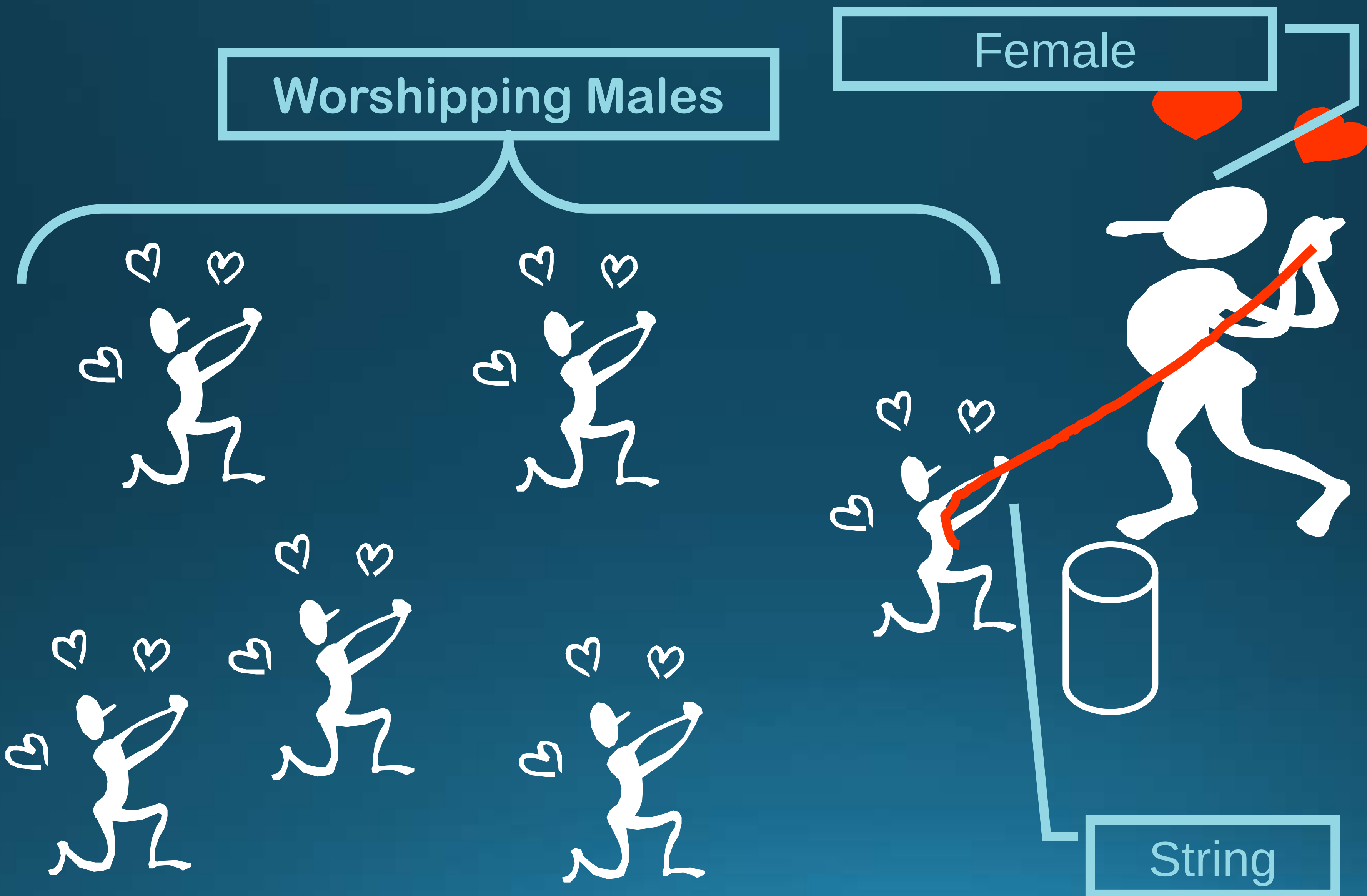
The non-bipartite case may not have any stable matching. In fact, this is the case for the example we just saw (Why?)

Lesson

Any proof of the existence of stable matching must contain a step that exploit bipartiteness

If we have a proof idea that works equally well in the bipartite and non-bipartite case, then our idea is not adequate to correctly establish the existence of stables matchings.

The Traditional Marriage Algorithm



The Traditional Marriage Algorithm

For each day some boy gets a “No” (& on first day) do:

Morning

- Each girl stands on her balcony
- Each boy proposes to his highest ranked girl whom he has not yet crossed off

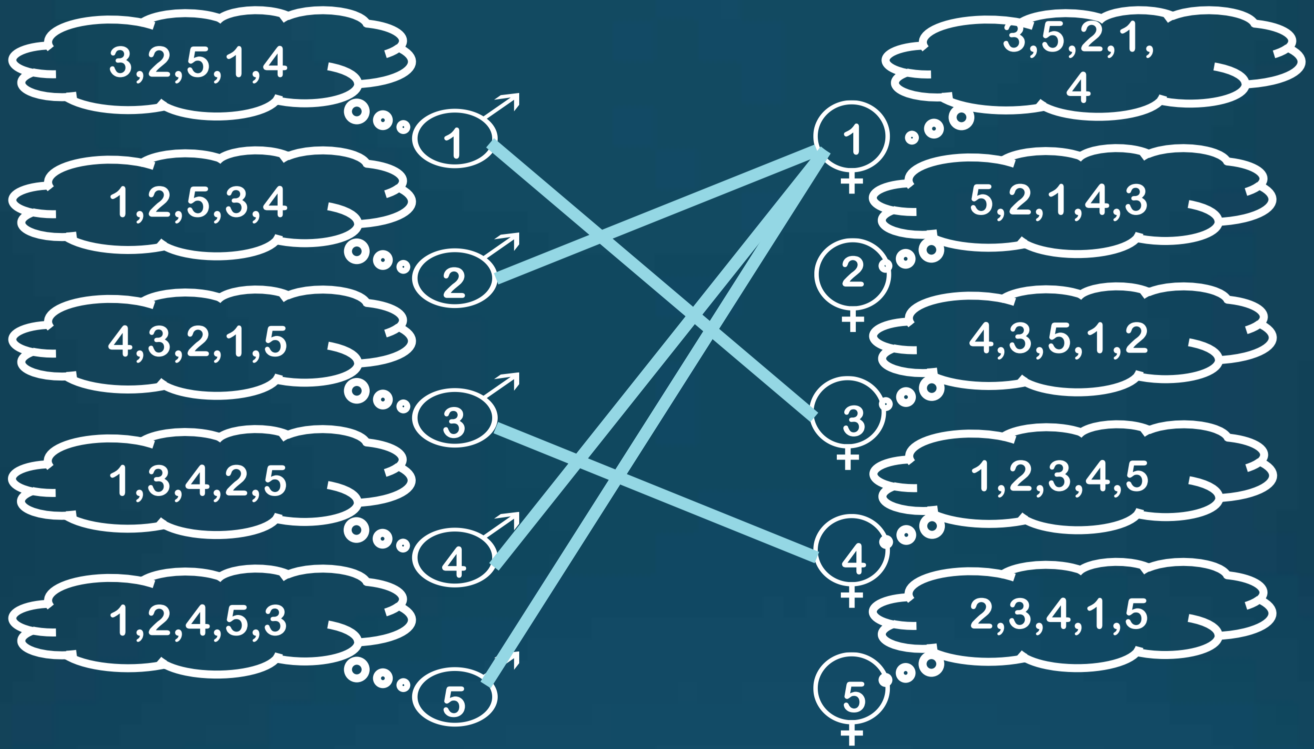
Afternoon (for girls with at least one suitor)

- To today’s best: “Maybe, return tomorrow”
- To any others: “No, I will never marry you”

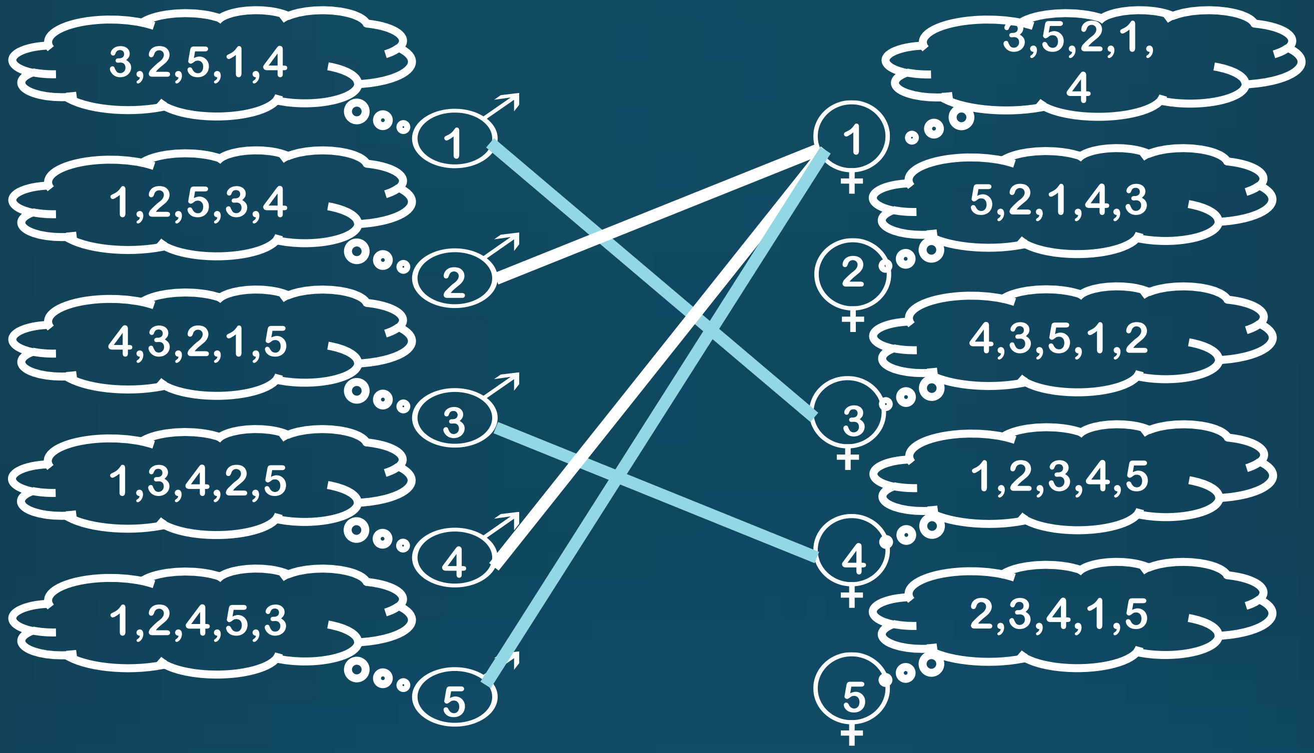
Evening

- Any rejected boy crosses the girl off his list

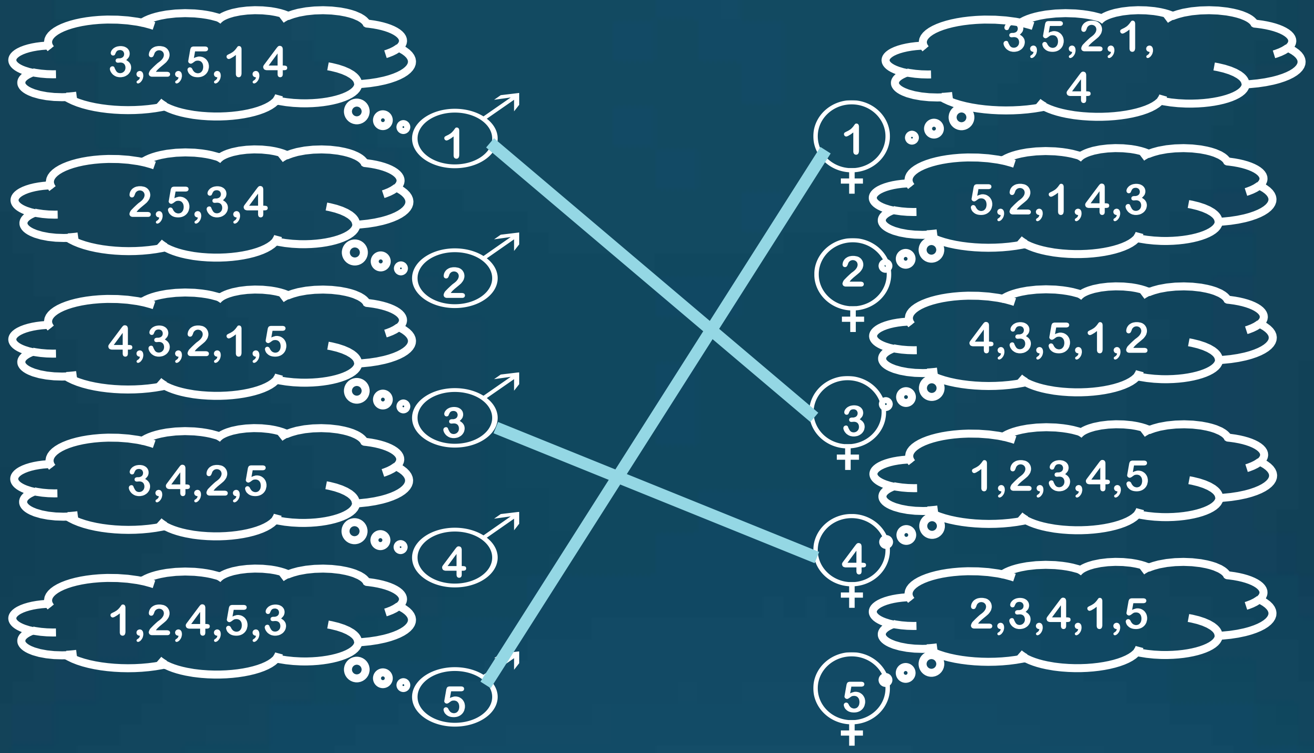
If every boy gets a “maybe,” each girl marries the boy to whom she just said “maybe”



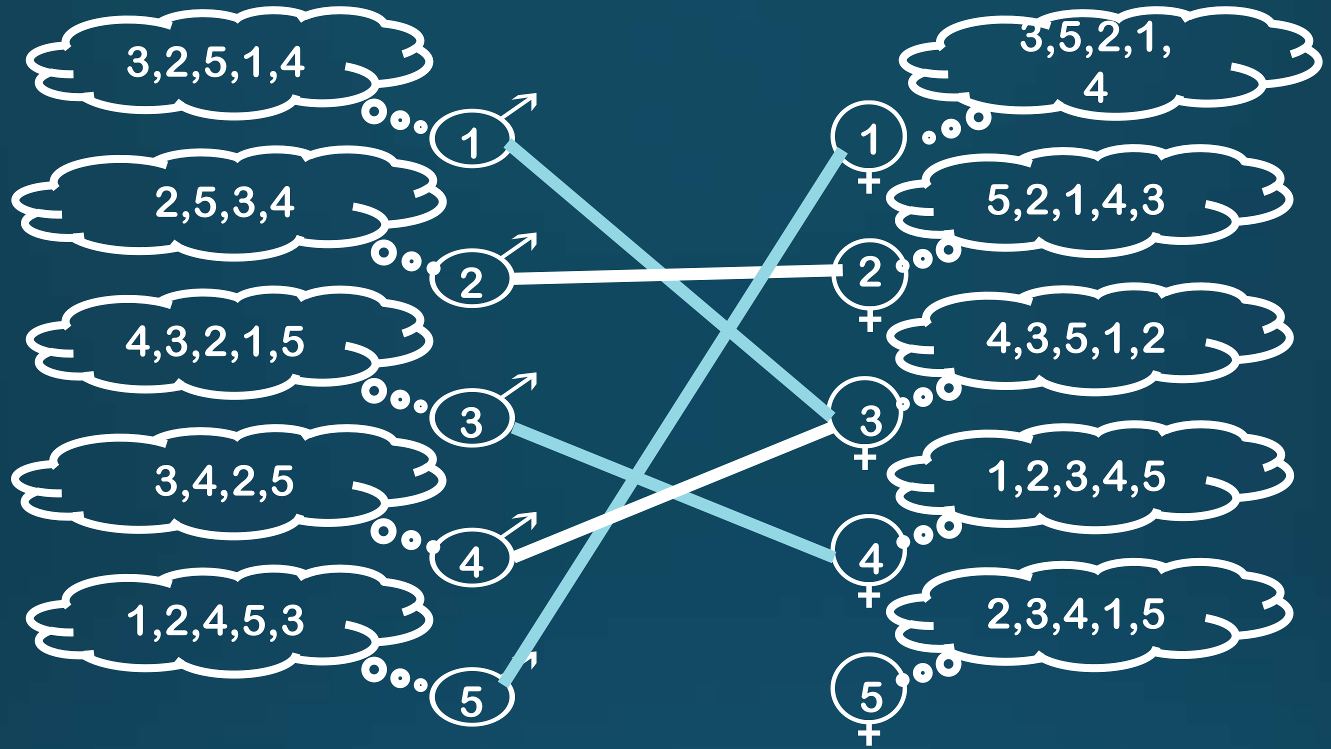
Traditional Marriage Algorithm Example



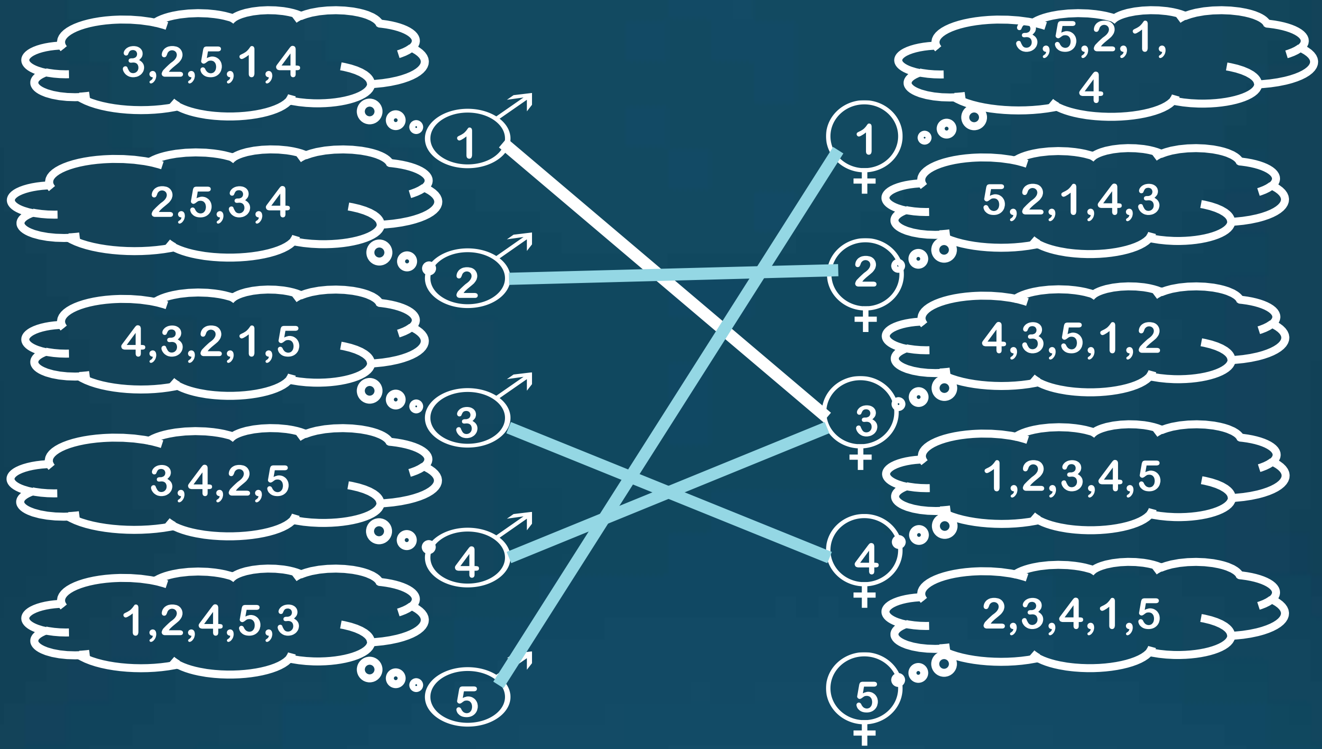
Traditional Marriage Algorithm Example



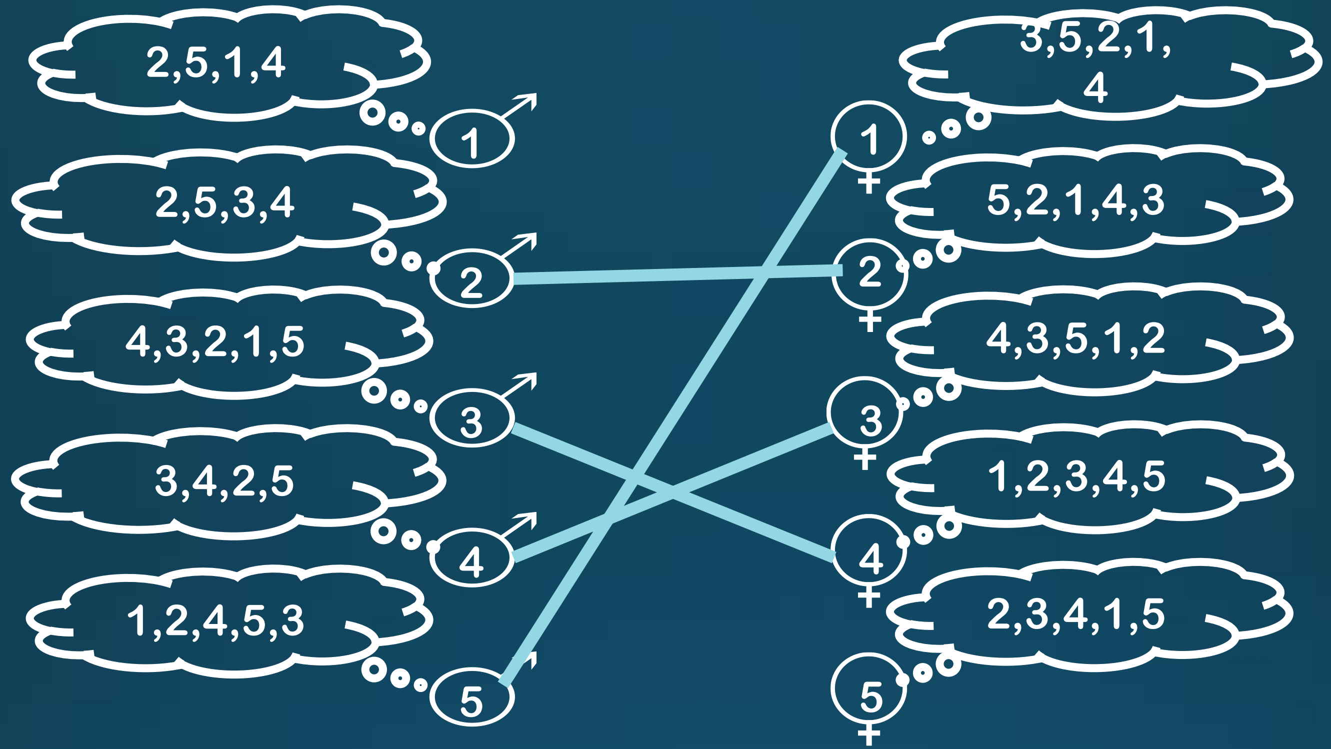
Traditional Marriage Algorithm Example



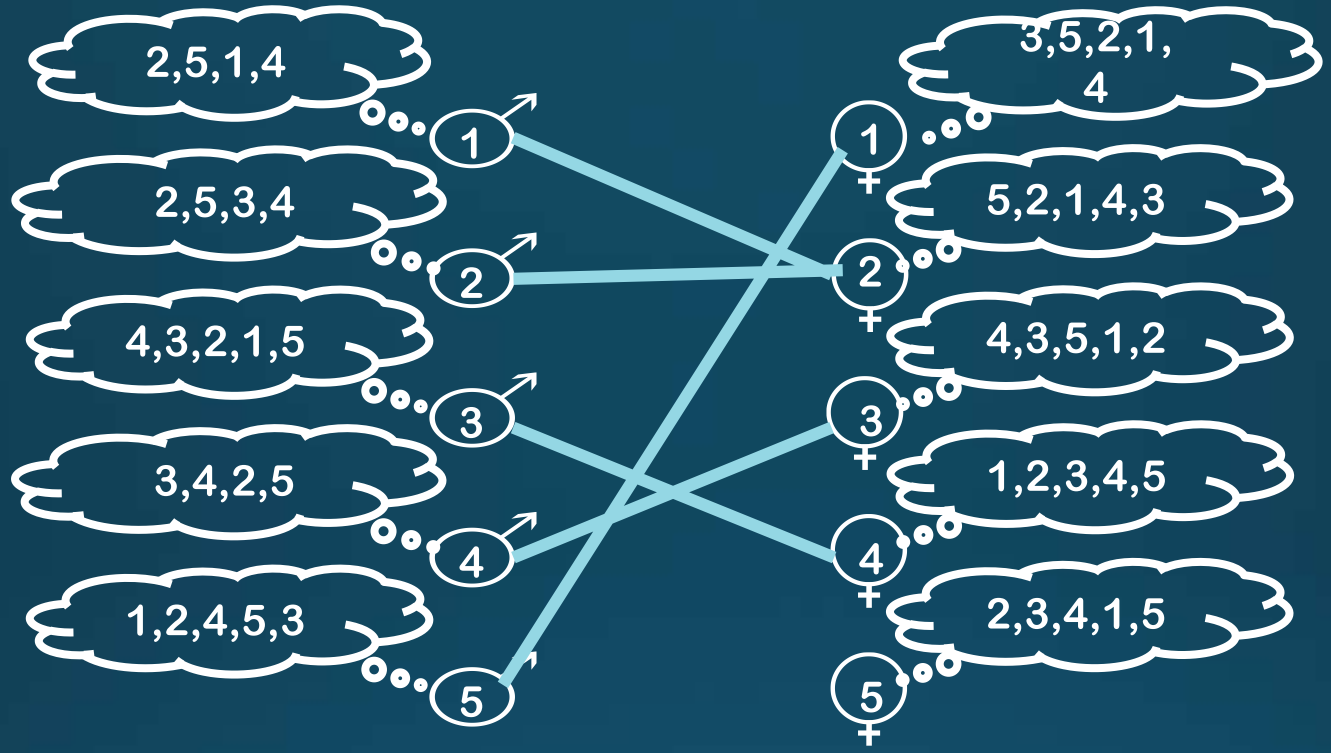
Traditional Marriage Algorithm Example



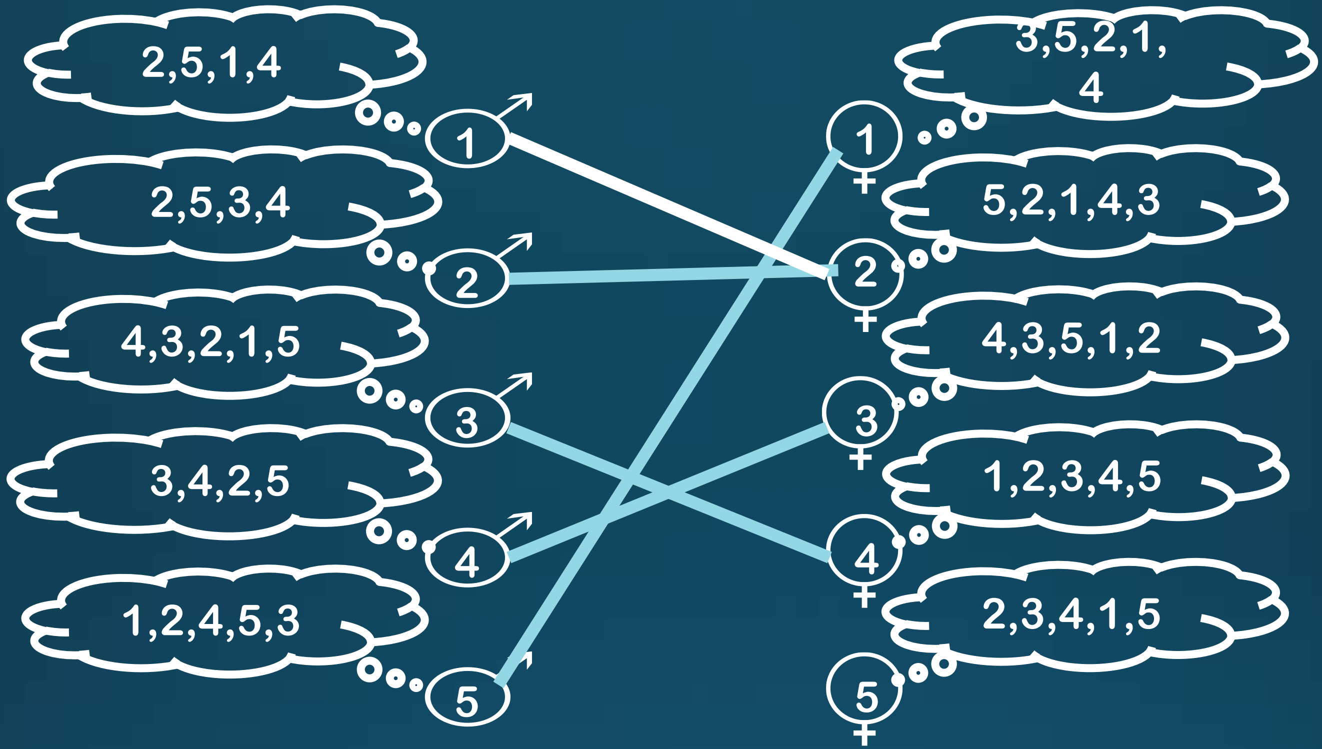
Traditional Marriage Algorithm Example



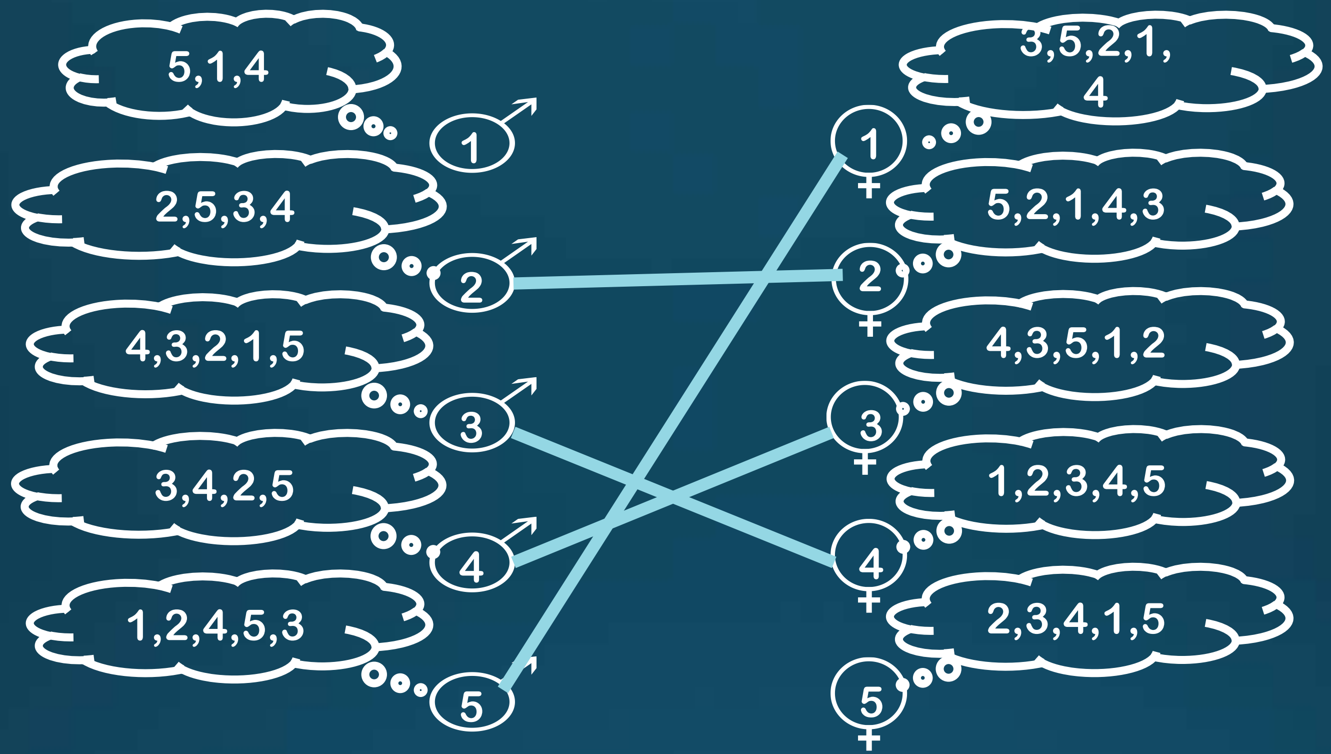
Traditional Marriage Algorithm Example



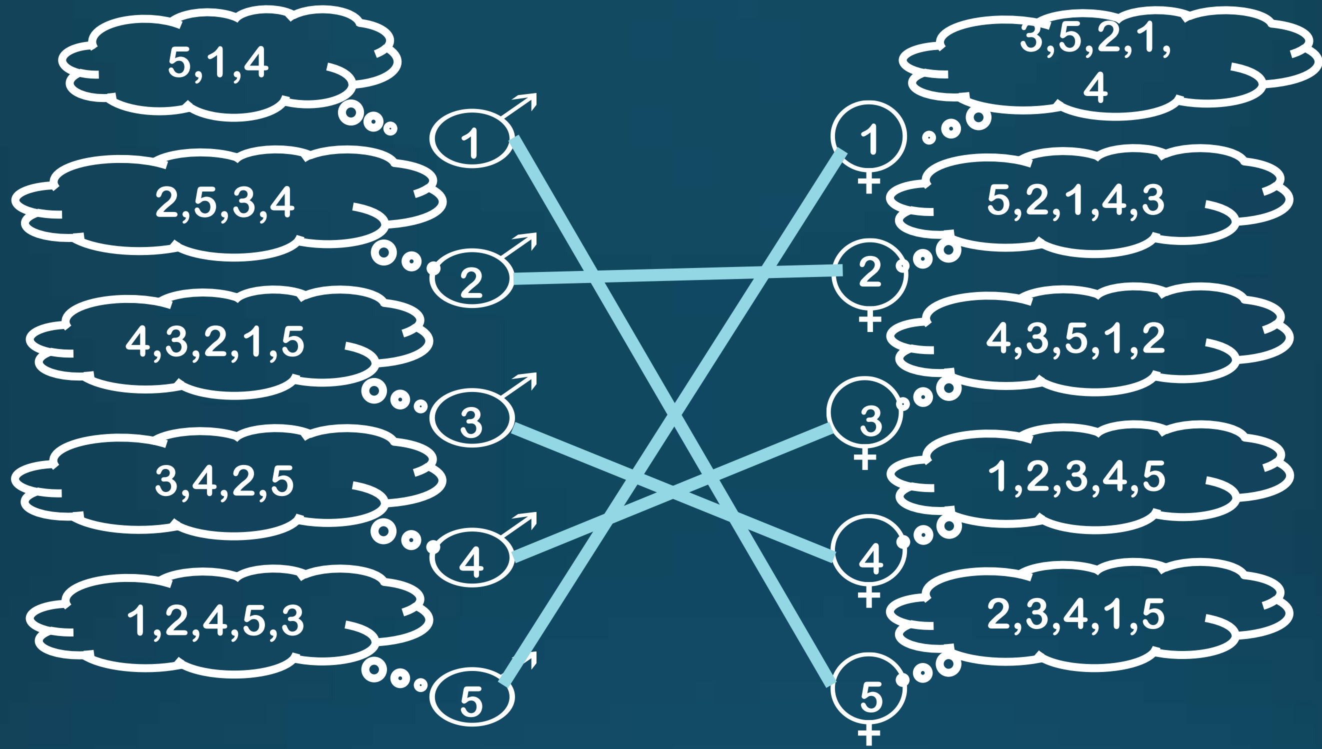
Traditional Marriage Algorithm Example



Traditional Marriage Algorithm Example

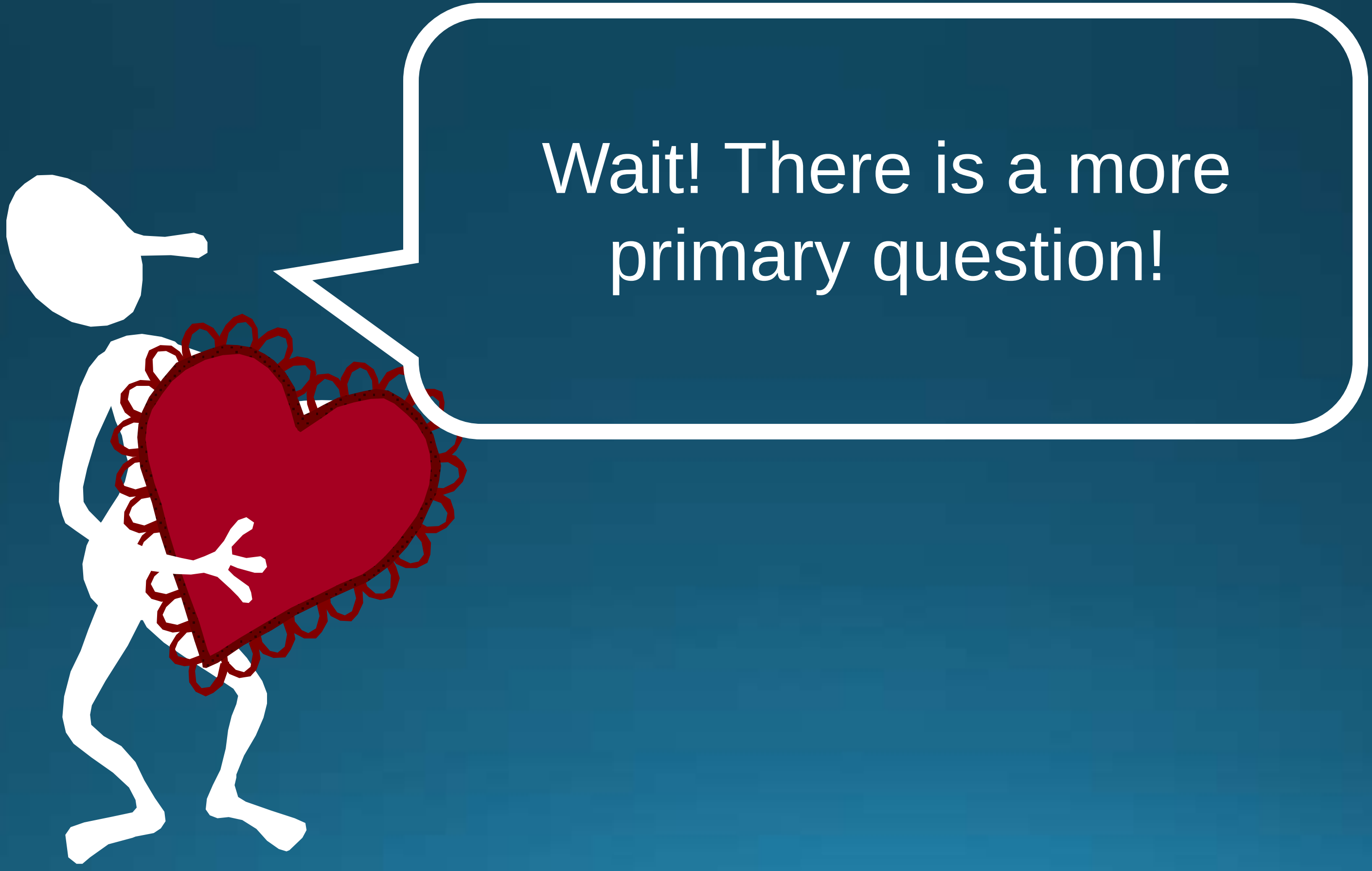


Traditional Marriage Algorithm Example



Traditional Marriage Algorithm Example

Does the Traditional Marriage Algorithm always produce a stable pairing?



Does TMA Always Terminate?

It might encounter a situation where the algorithm does not specify what to do next.

It might keep on going for an infinite number of days

Does TMA Always Terminate?

It might encounter a situation where the algorithm does not specify what to do next

this could happen if some boy crosses all names off his list!

we'll show it doesn't…

It might keep on going for an infinite number of days

cannot happen, every day some name is crossed off

Improvement Lemma:

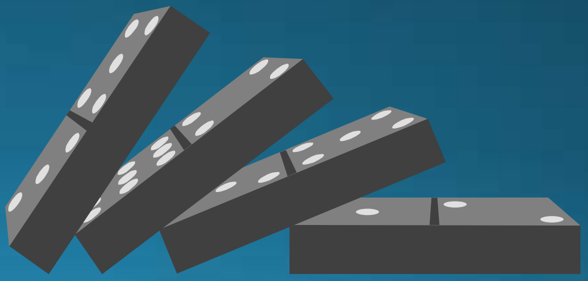
If a girl has a boy on a string, then she will always have someone at least as good on a string (or for a husband)

She would only let go of him in order to "maybe" someone better

She would only let go of that guy for someone even better

She would only let go of that guy for someone even better

AND SO ON…



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

Lemma: No boy can be rejected by all the girls

Proof (by contradiction):

Suppose boy b is rejected by all the girls

At that point:

- Each girl must have a suitor other than b
- (By Improvement Lemma, once a girl has a suitor she will always have at least one)

- The n girls have n suitors, and b is not among them. Thus, there are at least n+1 boys

Contradiction

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

A “master list” of all n of the boys’ lists starts with a total of $n \times n = n^2$ girls on it

Each day TMA doesn’t terminate, at least one boy gets a “No”, so at least one girl gets crossed off the master list

Therefore, the number of days is bounded by the original size of the master list

Great! We know that TMA terminates and produces a pairing

But is it stable?

Theorem: The pairing T produced by TMA is stable.

Let b and g be any couple in T, and b prefers g^* to g.
We claim that g^* prefers her husband to b.

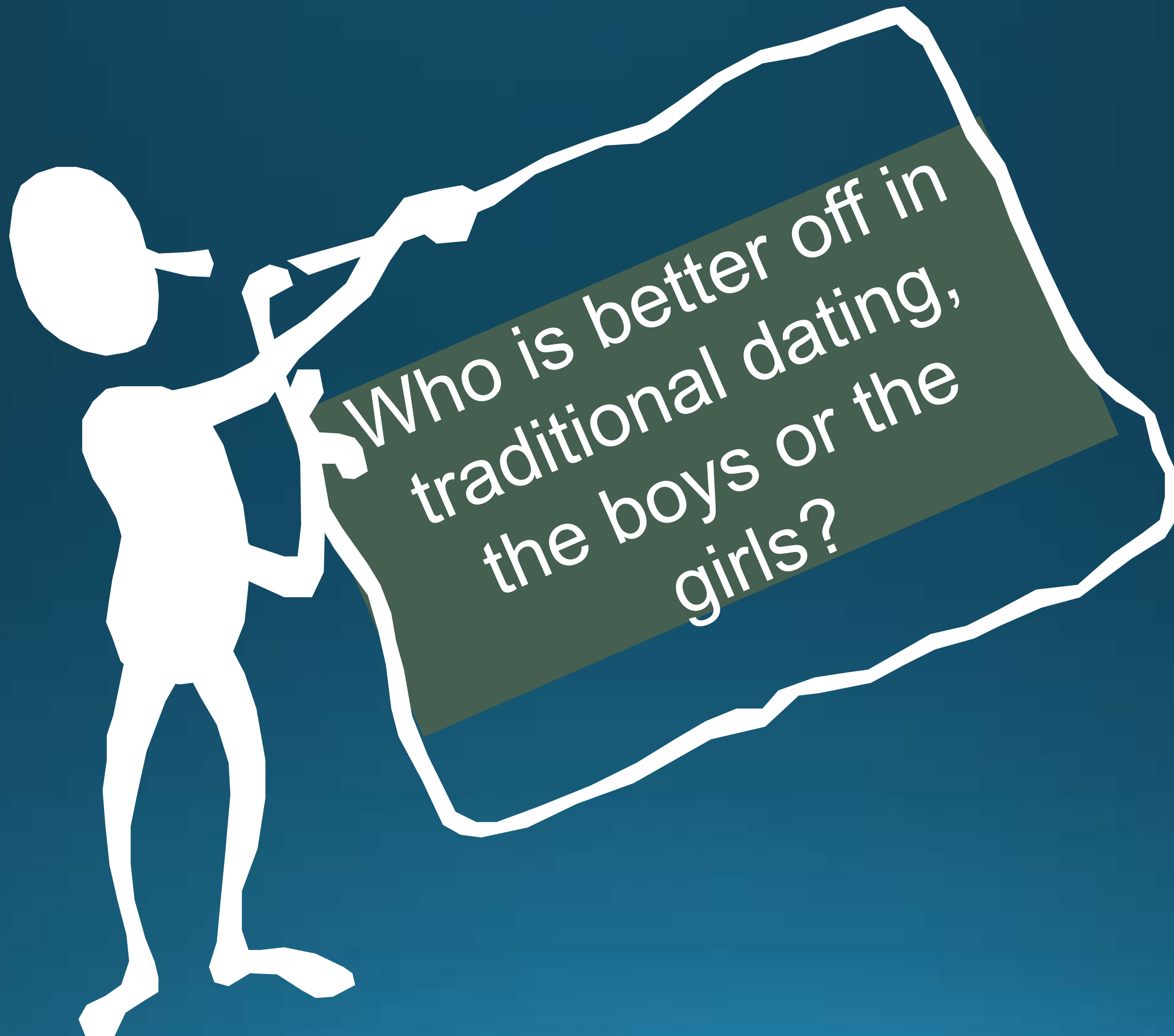
During TMA, b proposed to g^* before he proposed to g.

Hence, at some point g^* rejected b for someone she preferred to b.

By the Improvement lemma, the person she married was also preferable to b.

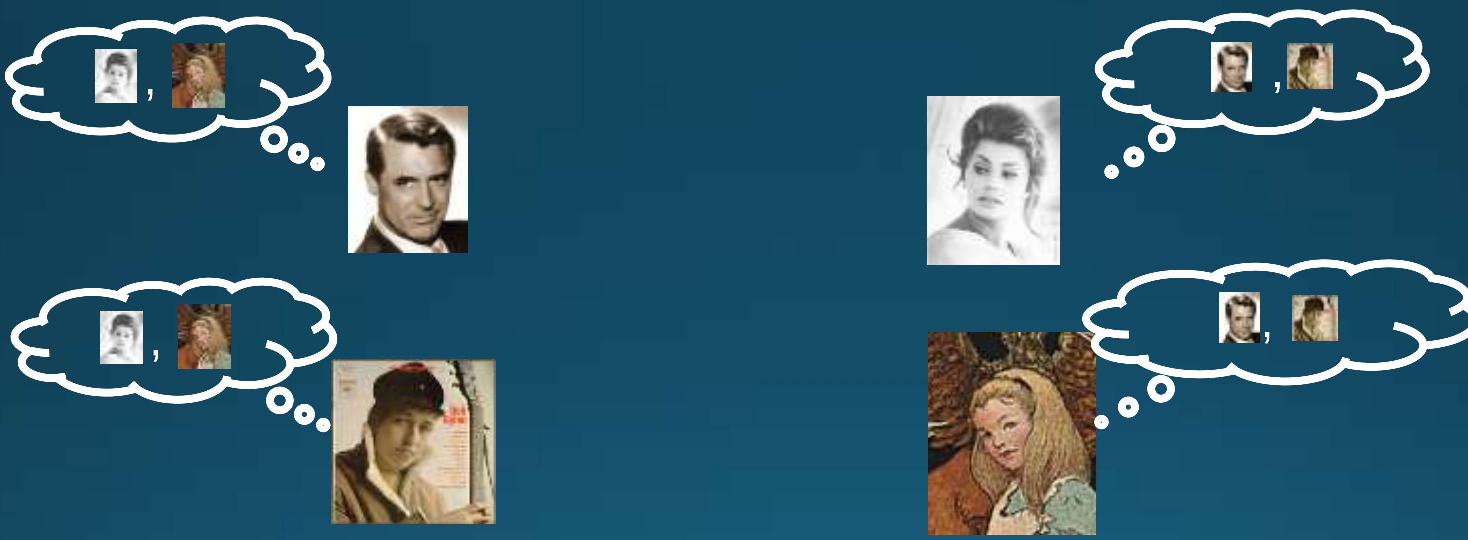
Thus, every boy will be rejected by any girl he prefers to his wife.

$\Rightarrow$ T is stable.


Who is better off in
traditional dating,
the boys or the
girls?

Forget TMA for a Moment…

How should we define
“the optimal girl for boy b”?



Flawed Attempt:
“The girl at the top of b’s list”

The Optimal Girl

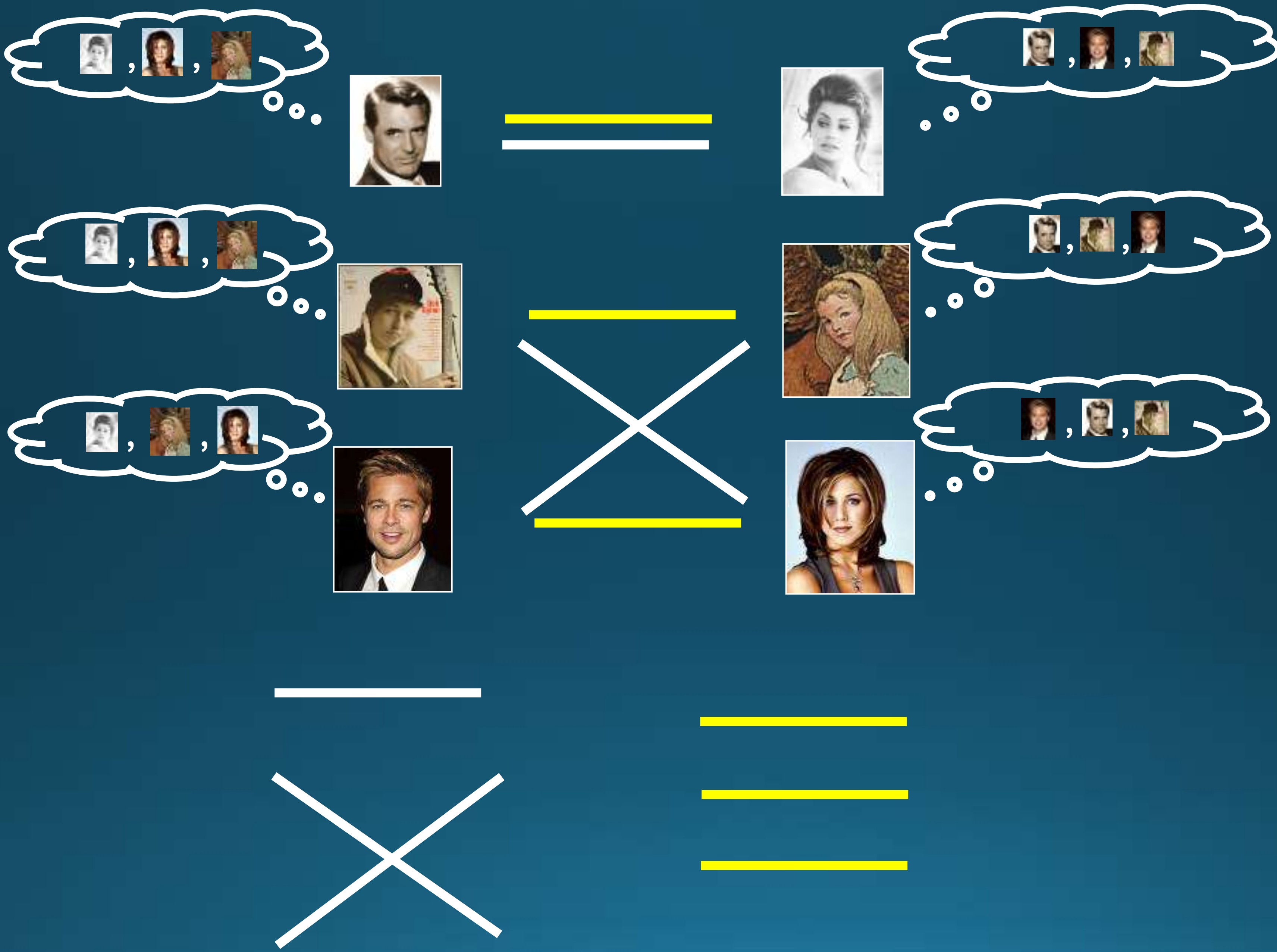
A boy's optimal girl is the highest ranked girl for whom there is some stable pairing in which the boy gets her

She is the best girl he can conceivably get in a stable world. Presumably, she might 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 conceivably 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 worlds for every boy simultaneously

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

not clear if either exists!

we'll show that both exist…

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

Opinion poll

The stable matching produced by the TMA algorithm is:

- Male-optimal
- Male-pessimal
- Female-optimal
- Female-pessimal
- Depends on the instance
- Beats me

(You may make multiple choices.)

The TMA has been used in the National Residency Matching Program (NMRP) since 1952

The TMA was first properly analyzed by Gale and Shapley, in a famous paper dating back to 1962:

D. Gale and L.S. Shapley,
“College Admissions and the Stability of Marriage,”
American Mathematical Monthly 69 (1962), pp. 9–14.

Stable marriage and its numerous variants remain an active topic of research in computer science.

The following very readable book covers many of the interesting developments since Gale and Shapley’s algorithm:

D. Gusfield and R.W. Irving, The Stable Marriage Problem: Structure and Algorithms, MIT Press, 1989.

The Mathematical Truth!

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

Theorem:

TMA produces a male-optimal pairing

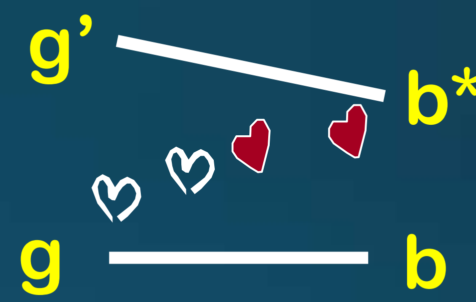
Suppose TMA
is not male-optimal

likes g
at least
as much as
his optimal
(not been
rejected by
optimal yet)

likes b*
more than b

consider the
first moment
in TMA when
some boy
is rejected by
his optimal girl

since g is b's optimal,
there is a stable matching S
where g and b are matched



g' at most
as good as
his optimal

likes b*
more than b

contradicts
stability of S!!!

Theorem: TMA produces a male-optimal pairing

Suppose, for a contradiction, that some boy gets rejected by his optimal girl during TMA

Let t be the earliest time at which this happened

At time t, boy b got rejected by his optimal girl g because she said “maybe” to a preferred b*

By the definition of t, b* had not yet been rejected by his optimal girl

Therefore, b* likes g at least as much as his optimal

Some boy b got rejected by his optimal girl g because she said "maybe" to a preferred b*. b* likes g at least as much as his optimal girl

Also, there must exist a stable pairing S in which b and g are married (by def. of "optimal")

b* wants g more than his wife in S:

g is at least as good as his optimal and he is not matched to g in stable pairing S

g wants b* more than her husband in S:

b is her husband in S and she rejects him for b* in TMA

Contradiction

Thm: The TMA pairing, T, is female-pessimal

In fact, we'll show a male-optimal pairing, (which T is) is female-pessimal

Let S be an arbitrary stable matching, and g any girl.

Let b be her mate in T = { …, (g,b), …}

Let b* be her mate in S = { …, (g,b*), (g',b), …}

b likes g better than his mate g' in S (because g is his optimal girl)

Since S is stable, g must like b* more than b (otherwise (g,b) will be a rouge couple in S)

So, b is the worst of all boys g can be paired with in a stable matching!

A graph is **planar** if it can be drawn in the plane without crossing edges

Examples of Planar Graphs

http://www.planarity.net

Faces

A planar graph splits the plane into disjoint faces

Euler’s Formula

If G is a connected planar graph
with n vertices, e edges and f faces,
then $n - e + f = 2$

Proof of Euler's Formula

For connected arbitrary planar graphs $n-e+f=2$

The proof is by induction.

Let's build up the graph by adding edges one at a time, always preserving the Euler formula.

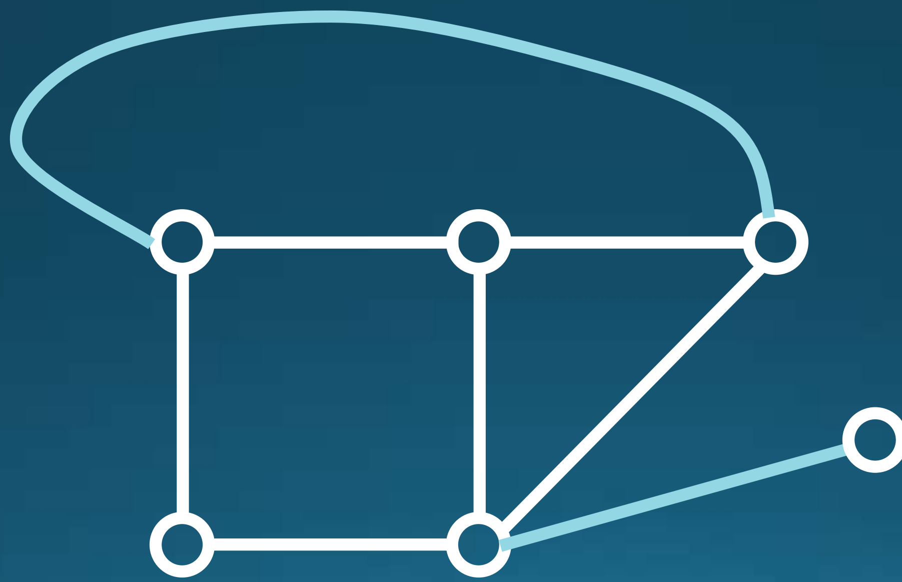
Start with a single edge and 2 vertices. $n=2$, $f=1$, $e=1$. Check.

Add the edges in an order so that what we've added so far is connected.

There are two cases to consider.

(1) The edge connects two vertices already there.
(2) The edge connects the current graph to a new vertex

In case (1) we add a new edge (e++) and we split one face in half (f++). So n-e+f is preserved.



In case (2) we add a new vertex (n++) and a new edge (e++). So again n-e+f is preserved.

Corollary 1: Let G be a simple connected planar graph with $n > 2$ vertices. G has at most $3n – 6$ edges

Proof: We already showed that under these conditions $2e \leq 6n-12$. Thus $e \leq 3n-6$. QED.

Note: This theorem is important because it shows that in a simple planar graph $e = O(n)$.

Corollary 1: Let G be a simple connected planar graph with $n > 2$ vertices. G has at most $3n – 6$ edges

Proof sketch: Each face has minimum 3 edges and each edge has maximum two faces. S**o** $3f \leq 2e$.

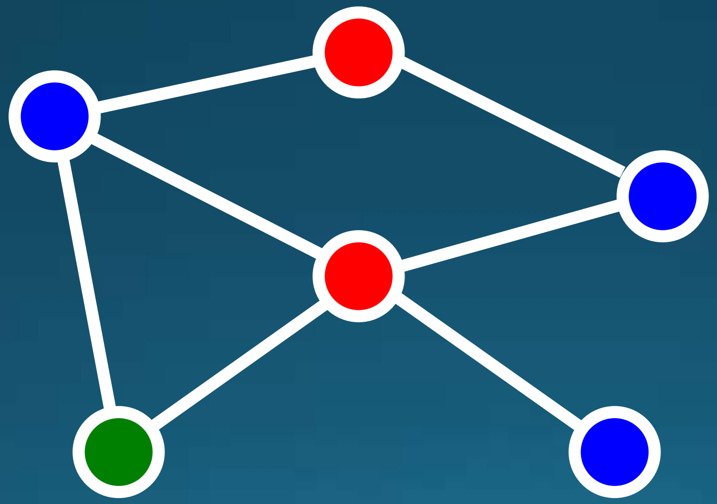
Euler: $n-e+f=2$

$\Rightarrow 3e = 3n+3f-6 \leq 3n+2e-6 \quad \Rightarrow \quad e \leq 3n-6$

Corollary 2: Let G be a simple connected planar graph. Then G has a vertex of degree at most 5.

Graph Coloring

A coloring of a graph is an assignment of a color to each vertex such that no neighboring vertices have the same color



Graph Coloring

Fundamental problem that arises surprisingly often

Register allocation: assign temporary variables to registers for scheduling instructions. Variables that interfere, or are simultaneously active, cannot be assigned to the same register

Assigning time slots for final exams: courses with common students can't be scheduled at the same time

Theorem: Every planar graph can be 6-colored

Proof (by induction):

Assume every planar graph with less than n vertices can be 6-colored. A base case of n<7 is trivial.

Assume G has n vertices

Since G is planar, it has some node v with degree at most 5.

Remove v and color remaining graph with 6 colors.

Now color v with a color not used among its at most 5 neighbors.

Theorem: Every planar graph can be 5-colored

Proof (by induction):

Assume every planar graph with less than n vertices can be 5-colored. A base case of n<6 is trivial.

Assume G has n vertices

Since G is planar, it has some node v with degree at most 5.

If deg(v)<5, Remove v and color remainin graph with 5 colors. Now color v with a color not used among its neighbors.

What if v has degree 5?

One can show that G – {v} admits a 5-coloring in which at most 4 colors are used amongst the neighbors of v.

One can extend this 5-coloring to G by using the 5'th color (that is unused among neighbors of v) to color v.

A computer-assisted proof of the 4-color theorem was discovered in 1976 by Appel and Haken of the University of Illinois.