2012 Nobel Prize in Economics

"The Prize concerns a central economic problem: how to match different agents as well as possible. For example, students have to be matched with schools, and donors of human organs with patients in need of a transplant. How can such matching be accomplished as efficiently as possible? What methods are beneficial to what groups? The prize rewards two scholars who have answered these questions on a journey from abstract theory on stable allocations to practical design of market institutions."

I will use the example of 1950's dating, but the relevant examples to keep in mind are things like matching medical residents with hospitals.

Stable Matching Problem

Goal. Given a set B and a set G, both of size n, find a "suitable" matching:

- Each b in B lists members of G in order of preference from favorite to least favorite
- Each g in G lists members of B in order of preference from favorite to least favorite.

B		G	
X:	A, B, C	A:	Y, X, Z
Y:	B, A, C	B:	X, Y, Z
Z:	A, B, C	C:	X, Y, Z

- What matching makes sense?
- Example: X-A, Y-C, Z-B

Stable Matching Problem

Perfect matching: every person is matched to exactly one person.

- Each b in B is matched to exactly one g in G.
- Each g in G is matched to exactly one b in B

Stability: no incentive for two people not matched with each other to try to switch to someone else

- In matching, an unmatched pair b-g' is **unstable** if both b and g' prefer each other to whoever they are actually matched to.
- Unstable pair b-g' could each improve by leaving their respective matches and "running off" together.

Stable matching: perfect matching with no unstable pairs.

Stable matching problem. Given the preference lists of members of B and G, find a stable matching **if one exists.**

Understanding the Solution

In this example

B

x: a, d, c

y: d, a, c

z: a, d, c

G:

a: y,x,z

c: x,y,z

d: x,y,z

There are two stable matchings:

x---a	x---b
y---d	y---a
z---c	z---c

Stable Matching Problem

Q. Do stable matchings always exist?

Not clear.

How would you go about checking?

B-proposing Algorithm [Gale-Shapley 1962]

Algorithm takes place over a series of days. Each day:

Morning:

Each g stands on her balcony. Each b that nobody has "tentatively accepted" stands under the balcony of the favorite g he has not yet ruled out and proposes that they match.

Afternoon:

Each g who has received a proposal tentatively accepts the proposal of her favorite proposer of those under the balcony and rejects the rest.

Evening:

Any b who was rejected, say by g, crosses g off his list.

Stopping condition: As soon as every b in B has either (a) been tentatively accepted or (b) have crossed everyone off their list, the process stops and all tentatively matched pairs are matched permanently.

Example

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

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

Day 1: X→ C, Y → A, Z → C

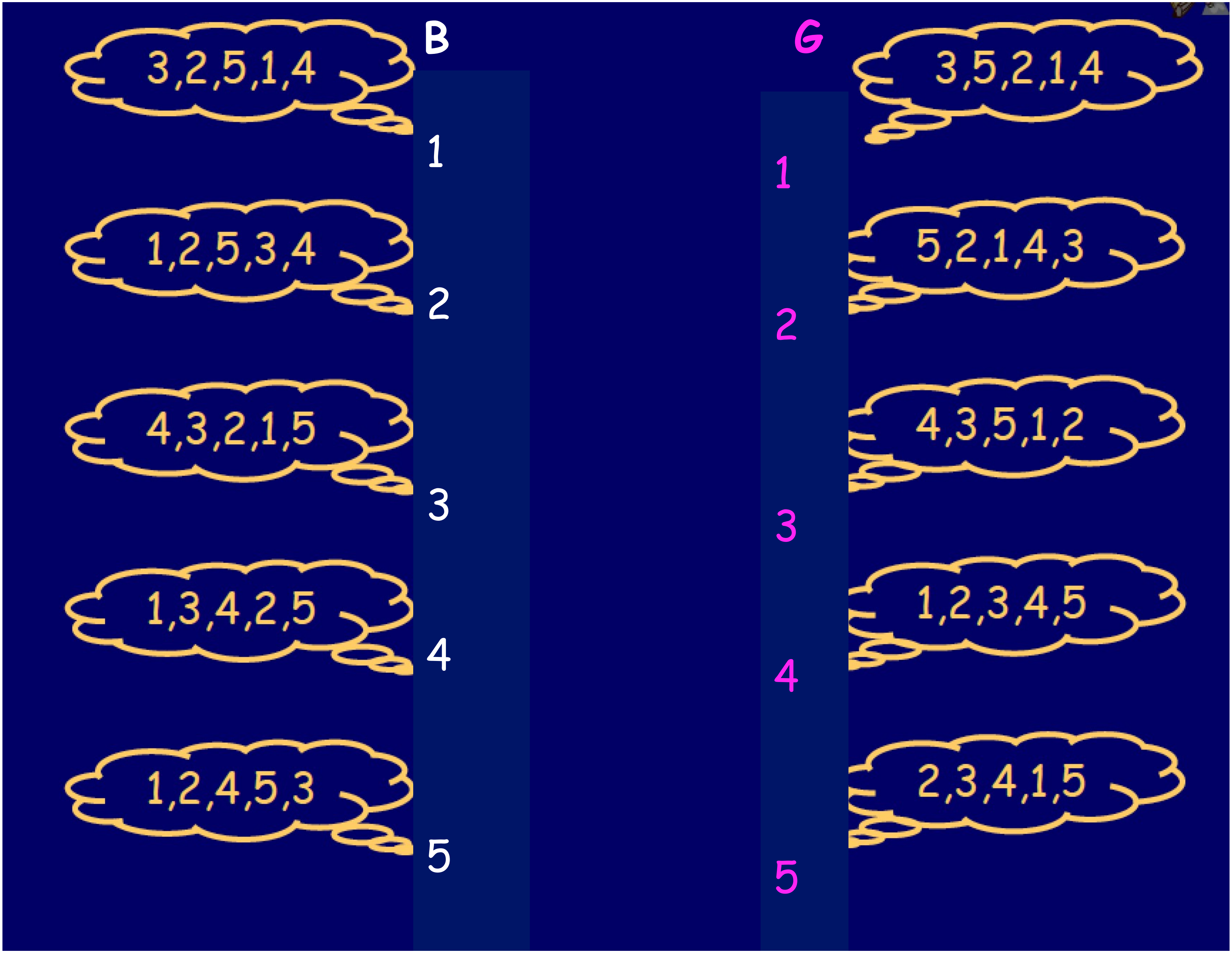
Tentative matches: X -- C, Y – A, Z crosses C off his list.

Day 2: X→ C, Y → A, Z → A

Tentative matches: X -- C, Y – A, Z crosses A off his list.

Day 3: X→ C, Y → A, Z → B

Done!


B
G
3,2,5,1,4
1
3,5,2,1,4
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

Some basic questions

Does the algorithm terminate or could it go on forever?

Yes, it always terminates!

Each step in which some b gets rejected, some *g* gets crossed off one of the lists.

That means that every b ends up matched or else is rejected by all *g* in *G*.

Is it possible for some boy be rejected by all the girls?

No!

Everybody is matched!

Observation 1. Once a g has received one proposal, she is guaranteed to be matched at the end; Also, the sequence of b's she is tentatively matched to get better and better (according to her preference list)!

Claim. This means that each girl ends up with her absolute favorite of all the boys who proposed to her.

Observation 2. Boys visit girls in decreasing order of preference.

If b is rejected by every g, then he has proposed to every g.

Which means that every girl ends up matched. Why?

Why does this mean that none of the n boys can be rejected by every girl?

The resulting matching is stable!!

Claim. No unstable pairs.

To see why not, suppose that b-g′ is an unstable pair.

Case 1: b never proposed to g′.

-Then b prefers his match to g′.

Case 2: b did propose to g′.

- Then g′ must have rejected him for someone she liked better.
- Her tentative matches got better and better
- Therefore g′ prefers her final match to b.

Summary

Stable matching problem. Given two sets B and G of equal size, and their preferences, find a stable matching if one exists.

B-proposing algorithm. Guarantees to find a stable matching no matter what the preference lists are!

Summary

Stable matching problem. Given two sets B and G of equal size, and their preferences, find a stable matching if one exists.

B-proposing algorithm. Guarantees to find a stable matching no matter what the preference lists are!

And does so efficiently!

There can be many stable matchings. Which one does this procedure find?

Understanding the Solution

Recall that for this example

B	**G:**
x: a, d, c	a: y,x,z
y: d, a, c	c: x,y,z
z: a, d, c	d: x,y,z

There are two stable matchings:

x---a	x---b
y---d	y---a
z---c	z---c

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. If so, which one?

Definition: *g* is an attainable match of b (and b is an attainable match of *g*) if **there exists** some stable matching in which they are matched.

Which are attainable matches of x in our previous example? y? z?

Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. If so, which one?

Definition: *g* is an attainable match of b (and b is an attainable match of *g*) if **there exists** some stable matching in which they are matched.

Claim: The outcome of the B-proposing algorithm is b-optimal!! Each b gets matched with his favorite attainable *g*.

The outcome of this procedure is simultaneously best for each and every b in B!!!!

Proof of B-optimality

Def. b is an attainable match of g if there exists some stable matching in which they are matched.

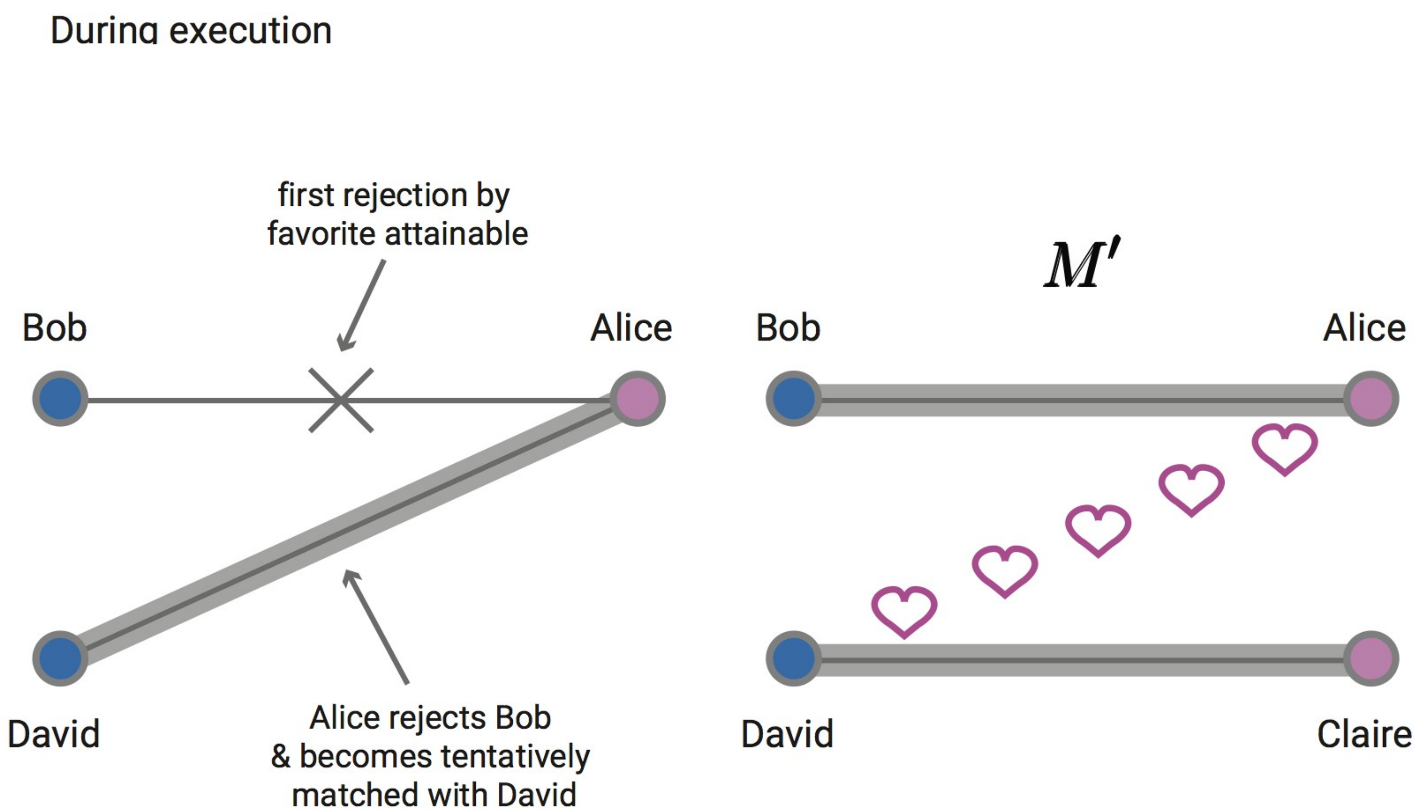
B-optimal assignment. Each b ends up with favorite attainable match!!!

Proof: Consider the first day that some b gets rejected by his favorite attainable match, say Bob is rejected by Alice for David.

Proof of B-optimality

Def. Boy b is an attainable date of girl g if there exists some stable matching in which they are matched.

B-optimal assignment. Each b ends up with favorite attainable match!!!



Summary so far

Stable matching problem. Given B and G of equal size , and their preferences, find a stable matching if one exists.

B-proposing algorithm. Guarantees to find a stable matching for any set of preference lists. And does so efficiently –number of steps is at most the sum of the lengths of the preference lists.

B-optimal assignment. Each b in B ends up with favorite attainable match!!!

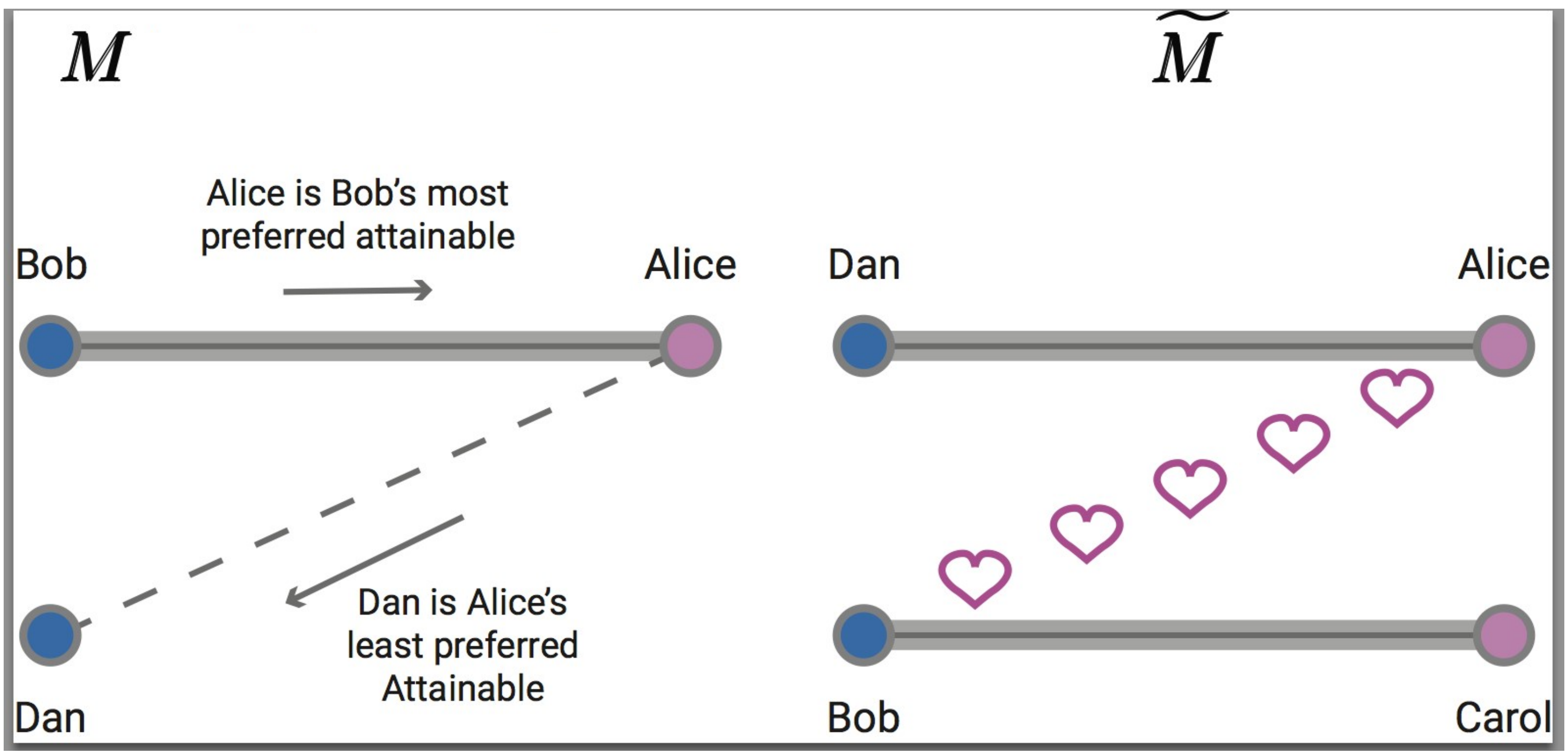
What about the members of G? How do they fare?

Not good....

Each g ends up with her **least** favorite (worst) attainable match!

Not good….

Bad for G!!! Each g ends up with her **least** favorite attainable match!



Summary

Stable matching problem. Given two sets B and G of equal size, and their preferences, find a stable matching if one exists.

B-proposing algorithm. Guarantees to find a stable matching for **any** set of preference lists. And does so efficiently –number of steps is at most the sum of the lengths of the preference lists.

B-optimal assignment. Each b ends up with favorite attainable match!!!

G-pessimal assignment. Each g ends up with her least favorite (worst) attainable match!!!