

Stable Matching, More Formally

Perfect matching:

- Each rider is paired with exactly one horse.
- Each horse is paired with exactly one rider.

Stability: no ability to exchange

an unmatched pair $r-h$ is **blocking** if they both prefer each other to current matches.

Stable matching: perfect matching with no blocking pairs.

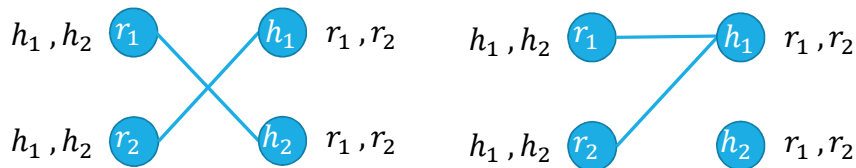
Stable Matching Problem

Given: the preference lists of n riders and n horses.

Find: a stable matching.

Try it!

Why are these not stable matchings?



Find a stable matching for this instance.

h_1, h_2, h_3 r_1 h_1 r_1, r_2, r_3

h_2, h_1, h_3 r_2 h_2 r_1, r_2, r_3

h_1, h_2, h_3 r_3 h_3 r_1, r_2, r_3

[Pollev.com/robbie](https://pollev.com/robbie)

Gale-Shapley Algorithm

Initially all r in R and h in H are free

While there is a free r

 Let h be highest on r 's list that r has not proposed to

 if h is free, then match (r, h)

 else // h is not free

 suppose (r', h) are matched

 if h prefers r to r'

 unmatch (r', h)

 match (r, h)

TODO List

Make sure you're on Ed! (check your spam folder for an invite, if not there send an email to Robbie).

Look at Homework 0 and 373 review materials.