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.

Claim 1: If r proposed to the last horse on their list, then all the horses are matched.

Try to prove this claim, i.e. clearly explain why it is true. You might want some of these observations:

Observation A: r's proposals get worse (for r).
Observation B: Once h is matched, h never becomes free again.
Observation C: h's partners cannot get worse (for h).

Hint: r must have been rejected a lot – what does that mean?

What data structures should you use?

Initially all r in R and h in H are free
While there is a free r — Need to maintain free r. What can insert and remove in $O(1)$ time?
  Let h be highest on r's list that r has not proposed to — Each r should know where it is on its list.
  if h is free, then match (r, h) — Maintain partial matching
  else //h is not free suppose (r', h) are matched
    if h prefers r to r′ — Given two riders, which horse is preferred?
      unmatch (r', h) — Maintain partial matching
      match (r, h)

Fill out the poll everywhere
Go to pollev.com/robbie and login
with your UW identity

Claim 4: The matching has no blocking pairs.

We want to prove a negative

  there is no blocking pair.

That's a good sign for proof by contradiction.

Suppose (for contradiction) that (r_1, h_1) and (r_2, h_2) are matched, but

  r_1 prefers h_2 to h_1 and

  h_2 prefers r_1 to r_2