

recap

Propose and Reject Algorithm

Job-Proposes Scenario:

- ① each job proposes to its favorite candidate that hasn't rejected it yet
- ①.5 if each candidate has a proposal, we're done.
- ② each candidate reviews her proposals, puts her favorite one "on a string", and explicitly rejects the rest.
- ③ each job crosses off the candidates who have rejected that job.

Precise Definitions (important!)



Stable. no rogue couples

rogue couple. job A and candidate B who BOTH prefer each other to their current match

Stable matching problem. MUST have 2 distinct categories of things

Stable matching instance. list of preferences of jobs and candidates (sometimes we look @ this "instance" and analyze both stable & unstable pairings.

[job] optimal matching. each job is paired with its highest preferred candidate that it can be paired with and still have a stable matching

optimal. "best valid partner"

proposal. every job proposes to its preferred candidate until explicitly rejected.



recap Proofs / Lemmas (assuming job-propose for this)

Improvement Lemma. once candidate C gets a proposal J , everyday she gets a proposal that is at least as good as J .

Well-Ordering Principle. any nonempty subset of \mathbb{N} has a "smallest element".

- what do we need this for? finding a "smallest counterexample", doing induction in general. there needs to be a **first domino** for us to push over!

P&R always halts. if it didn't halt, J rejected n times \rightarrow each of n candidates has a job better than $J \rightarrow$ more than n jobs.

P&R \rightarrow stable matching. final couple $(J, C) \rightarrow$ suppose J wanted C^* . then J must have proposed to C^* first. C^* only rejected J if she had something better, so she does NOT want to switch. no rogue couple.

- if C wanted J^* — this just breaks improvement lemma directly.

Job Propose is Job optimal (thru P&R). assume P&R gives non-job optimal matching, and there's some other job-optimal matching T . prove that T is actually not stable.



key takeaways

- Stable Marriage induction: usually induct on what day you're on.
 - base case: current day
- for contradiction proofs, try to tie it back to the improvement lemma
- most of these proofs are an exercise in definitions.
- **Well-ordering principle**: "day k is the first day when X happens"
- remember that any candidate prefers to have a job than no job
 - getting at least one proposal \Rightarrow have a job on a string $\ddot{\text{c}}$

Shoot your shot, friends, it works out in your favor $\ddot{\text{c}}$ go apply to that job!

thanks for coming! help me help you \rightarrow <https://www.tinyurl.com/aishani-sp21-fb>