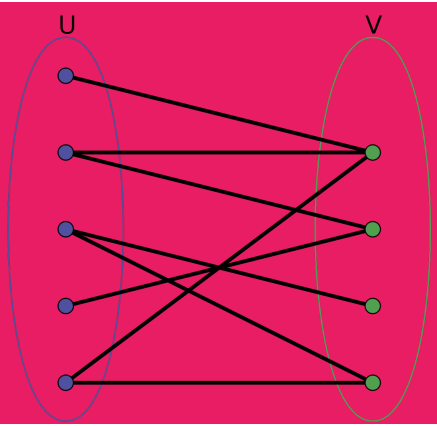
The Stable Matching Problem

Week 6 AI Inspire Winter 2020

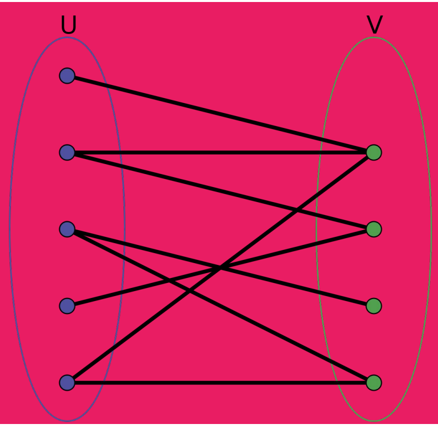
Tentative schedule of remaining sessions

1) Week 6 (Today) ⇒ Stable Matching
2) Week 7 ⇒ P vs NP
3) Week 8 ⇒ P vs NP part 2
4) Week 9 ⇒ other impo graph algos (some don’t fit under larger umbrella) we haven’t covered in past (or even variations of shortest path, etc.)
5) Week 10 ⇒ Summary + Ongoing research + future trends

Spring session: Quantum computing

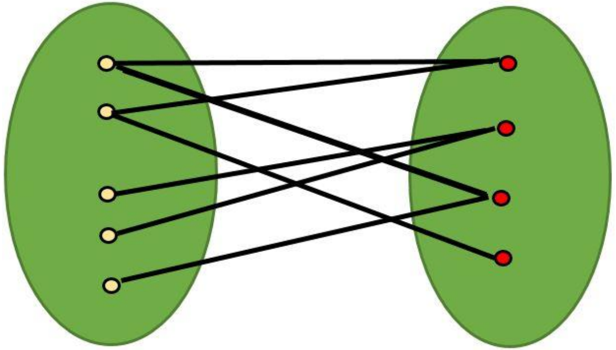


Introduction



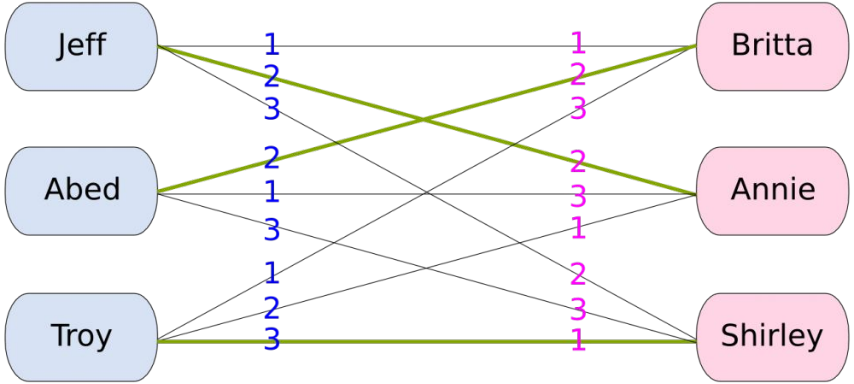
Applications

- Professors to courses
- Employees to tasks
- Stable marriage problem (men to women)
- College admissions process
- Many other similar problem statements



Stable marriage problem

- N men and N women
- Men and women have set of preferences
- Want to match the men and women in pairs



Some vocabulary

- **Matching S** = set of pairs s.t. each member of M and each member of W appears in AT MOST 1 PAIR
- **Perfect matching S** = set of pairs s.t. Each member of M and each member of W appears in EXACTLY 1 PAIR
- **Stable matching S** = type of perfect matching with NO INSTABILITY (adding preferences or ranking)
- **Instability** = pair (m, w’) is an instability when both m and w’ prefer each other in their rankings but are not a pair in S

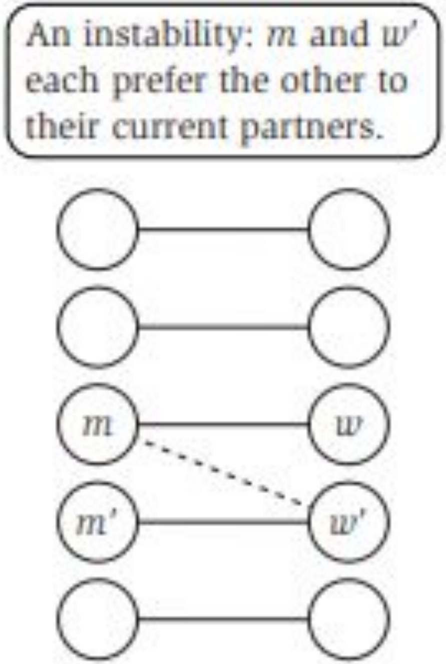


Figure 1.1 Perfect matching S with instability (m, w').

Image credits = Algorithm Design by Kleinberg & Tardos

Interesting things to think about

- Does there exist a stable matching for every bipartite graph?
- When there is a stable matching for a bipartite graph, can we efficiently construct one?

Scenarios

In both of these scenarios ⇒ M = {m, m'} and W = {w, w'}.

Goal ⇒ Given following scenarios with preferences, construct STABLE matching

1) Scenario 1 - only 1 stable matching
   a) m ⇒ {w, w'} AND m' ⇒ {w, w'}
   b) w ⇒ {m, m'} AND w' ⇒ {m, m'}
2) Scenario 2 - 2 stable matching possibilities
   a) m ⇒ {w, w'} AND m' ⇒ {w', w}
   b) w ⇒ {m', m} AND w' ⇒ {m, m'}

Gale-Shapley (GS) Algorithm

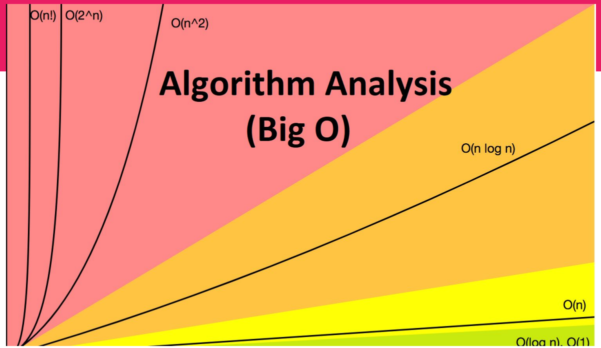
Pseudocode

- Initialization stage: everyone is unmarried
- Unmarried man m chooses woman w who's highest on his list to propose to ⇒ not necessary that (m, w) is part of stable matching = engagement
  - Because woman may override and accept somebody's offer or she also may not have any other options
- Continue with next unmarried man
  - If w is free ⇒ engagement, else w checks if need to override previous engagement with new man by looking at her preference list
- Algorithm is over when engagements are COMPLETE and final

```
Initially all m ∈ M and w ∈ W are free
While there is a man m who is free and hasn't proposed to
every woman
   Choose such a man m
   Let w be the highest-ranked woman in m's preference list
      to whom m has not yet proposed
   If w is free then
      (m, w) become engaged
   Else w is currently engaged to m′
      If w prefers m′ to m then
         m remains free
      Else w prefers m to m′
         (m, w) become engaged
         m′ becomes free
      Endif
   Endif
Endwhile
Return the set S of engaged pairs
```

Image credits = Algorithm Design by Kleinberg & Tardos

Algorithm Analysis



Important lemmas

1) Woman w will remain engaged till the end ever since she’s first engaged
   a) Beginning ⇒ man proposes to her ⇒ she’s engaged, later iterations w can decide whether to switch her partner or not
2) As time progresses, man m proposes to worse women on his list (in terms of his preference list)
   a) Man m will always go with highest ranked one and when he can’t go with highest, he will try proposing with ones who are lower ranked on his list

More important lemmas

1) Gale-shapley algorithm will end in runtime proportional to n^2 (n^2 loops/iterations of while loop)
    a) What step helps us get closer to termination?
    b) Each iteration ⇒ man m proposes to a new woman w (once and only once)
        i) n^2 total possible pairs of men and women
        ii) Max of n^2 possible proposals so worst case is proportional to n^2

Proving S (result) is a PERFECT MATCHING

Perfect matching = NO FREE MAN. PROOF =

1) If m is free during anytime of the algo, he hasn’t proposed to a woman w.
   a) Assume by contradiction that m has proposed to every woman and he’s still free. Then, all n women are engaged. This implies that there must be n men also engaged because of the pair property. Thus, we’ve proved the claim.
2) Result - set S - is a perfect matching
   a) Let us assume algorithm ends with a free man m. But, we know that according to algorithm, m would have proposed to every woman to see if he can get engaged. This contradicts previous lemma, which states that there can’t exist a free man who proposed to every woman since he hasn’t proposed to a woman w.

Proving S results in a STABLE matching

TRY PROVING THIS ON YOUR OWN FIRST

USE PROOF BY CONTRADICTION (take actual case)

http://www.cs.sjtu.edu.cn/~jiangli/teaching/CS222/files/materials/Algorithm%20Design.pdf

Summary of problem Itself: https://www.youtube.com/watch?v=RE5PmdGNgjO

Introduction to next session:
https://www.youtube.com/watch?v=EHp4FPyajKQ
AND
https://www.youtube.com/watch?v=0Y41QYPI8cw&t=1s