CSE 421 Algorithms

Richard Anderson

Lecture 1

Course Introduction

- Instructor
  - Richard Anderson, anderson@cs.washington.edu
- Teaching Assistant
  - Yiannis Giotas, giotas@cs.washington.edu

All of Computer Science is the Study of Algorithms

Mechanics

- It's on the web
- Weekly homework
- Midterm
- Final exam

- Subscribe to the mailing list

Text book

- Algorithm Design
- Jon Kleinberg, Eva Tardos

- Read Chapters 1 & 2

How to study algorithms

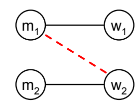
- Zoology
- Mine is faster than yours is
- Algorithmic ideas
  - Where algorithms apply
  - What makes an algorithm work
  - Algorithmic thinking

Introductory Problem: Stable Matching

- Setting:
  - Assign TAs to Instructors
  - Avoid having TAs and Instructors wanting changes
    - E.g., Prof A. would rather have student X than her current TA, and student X would rather work for Prof A. than his current instructor.

Formal notions

- Perfect matching
- Ranked preference lists
- Stability



Examples

- m_1: w_1 w_2
- m_2: w_2 w_1
- w_1: m_1 m_2
- w_2: m_2 m_1

- m_1: w_1 w_2
- m_2: w_1 w_2
- w_1: m_1 m_2
- w_2: m_1 m_2

Examples

- m_1: w_1 w_2
- m_2: w_2 w_1
- w_1: m_2 m_1
- w_2: m_1 m_2

Intuitive Idea for an Algorithm

- m proposes to w
  - If w is unmatched, w accepts
  - If w is matched to m_2
    - If w prefers m to m_2, w accepts
    - If w prefers m_2 to m, w rejects
- Unmatched m proposes to highest w on its preference list

Algorithm

```
Initially all m in M and w in W are free
While there is a free m
        w highest on m's list that m has not proposed to
        if w is free, then match (m, w)
        else
                suppose (m2, w) is matched
                if w prefers m to m2
                        unmatch (m2, w)
                        match (m, w)
```

Does this work?

- Does it terminate?
- Is the result a stable matching?

- Begin by identifying invariants and measures of progress
  - m's proposals get worse
  - Once w is matched, w stays matched
  - w's partners get better

Claim: The algorithms stops in at most n^2 steps

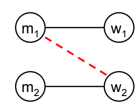
- Why?

The algorithm terminates with a perfect matching

- Why?

The resulting matching is stable

- Suppose
  - m_1 prefers w_2 to w_1
  - w_2 prefers m_1 to m_2



- How could this happen?