# CS 580: Algorithm Design and Analysis

Jeremiah Blocki
Purdue University
Spring 2019

# Algorithm Design, Analysis and Implementation

- Algorithm Design
  - **Key Techniques to design efficient algorithms:** Greedy, Divide & Conquer, Dynamic Programming etc…
  - **Involves:** Critical thinking + problem solving
- Algorithm Analysis
  - Quantify resources necessary to run the algorithm
    - Running Time (#Steps), Space (Required Memory), etc…
  - Prove that the output is correct.
    - Requires clear definition of desired input/output behavior
- Implementation: Specify the steps clearly and precisely in pseudocode
  - Should be straightforward to translate pseudocode to C/Java/Python etc…
  - No programming assignments
- Remark: Clear pseudocode is strongly preferred to obfuscated C/Java/Python etc… code in homework solutions

# Administrative Stuff

Lectures. Jeremiah Blocki
- Tuesday/Thursday 3PM - 4:15, FNY B124

Office Hours: Wed/Fri 11am-noon
- Lawson 1165
- Google Hangouts (EPE Students)

TAs.
- Akash Kumar (Office Hours: Monday 10 AM-Noon @ LWSN 3133)
- Hamidreza Amini Khorasgani (Office Hours: Tue/Thu 1-2PM @ HAAS G50)

Prereq. Mathematical maturity. Undergraduate algorithms (e.g., CS 381).

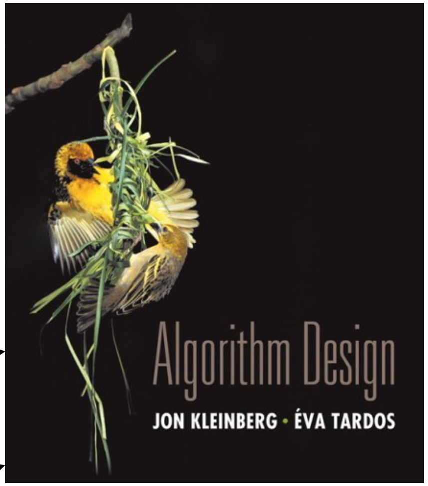
Textbook. *Algorithm Design* by Jon Kleinberg and Éva Tardos.

Course web site:
www.cs.purdue.edu/homes/jblocki/courses/580_Spring19/

# Course Readings

## Schedule (subject to change)

| | |
|---|---|
| Jan. 8 | Lecture 1 (Slides). Introduction. The stable matching problem. [Read Chapter 1 of Algorithm Design]. **Homework 0** |
| Jan. 10 | Lecture 2 (Slides). Asymptotic complexity. [Read Chapter 2] |
| Jan. 15 | Lecture 3 (Slides). Basic graph algorithms [Read Chapter 3] **Hw 1 Released .tex** |
| Jan. 17 | Lecture 4 (Slides). Lecture 4. Graph Algorithms Continued [Read Chapter 4.1-4.3] |
| Jan. 22 | Lecture 5 (Slides). Greedy Algorithms, Shortest Path [Read Chapter 4.4] |
| Jan. 24 | Lecture 6. (Slides) Cache Policies, Minimum Spanning Tree [Read Chapter 4.5-4.6] **Homework 1 Due, Hw 2 Released .tex** |




https://www.cs.purdue.edu/homes/jblocki/courses/580_Spring19/

# Grading

- 20% for homework
- 20% for the midterm 1 (Feb 21. Evening Exam)
- 20% for the midterm 2 (April 4. Evening Exam)
- 35% for the final
- 5% for class participation.
  - EPE Students: Piazza Discussion
  - On Campus: Participation during lecture + piazza discussion

https://piazza.com/purdue/spring2019/cs580

# Homework

- 20% for homework
  - Weekly/Biweekly problem sets
  - Turn in online on Gradescope
    - Entry Code: **97D2ER**
  - Typed Solutions (LaTeX, Word)
  - Discussion on Piazza

# Homework Policy

- Difficult Question?
  - Partial Credit (15%) for admitting "I don't know the answer."
  - No credit for bad answer
- Collaboration
  - You may discuss homeworks with classmates and course staff
    - Office Hours
    - Discussion on piazza
  - However…
    - You must write down your own solutions
    - You must completely understand your solution!
    - You must acknowledge your collaborators!
    - You may not search online for solutions (Course Hero, Stack Overflow etc…)

# Homework Policy

- Late Work?
  - 0.01 to 24 hours late (10% penalty)
  - 24 to 48 hours late (25% penalty)
  - > 48 minutes late (no credit)
- Exceptions for extreme circumstances
  - Serious Illness
  - Bereavement
- Re-grading
  - Must be done within two weeks of the day the work is returned to class.
  - Entire assignment/exam may be re-graded

# Exams

- Most Significant Part of Your Grade
  - 20% for the midterm 1 (Feb 20. Evening Exam)
  - 20% for the midterm 2 (April 3. Evening Exam)
  - 35% for the final (TBD. April 29-May 4th)
    - Plan to remain on campus through May 4th until final exam has been scheduled

- Allowed to bring double sided index card (3x5 inches) with your own notes
  - No electronics

- Disabilities Requiring Special Accommodations
  - Speak with me within the first three (3) weeks of the semester.
  - **Note:** We cannot arrange special accommodations without confirmation from the Disability Resource Center here at Purdue (http://www.purdue.edu/drc)

# Students with Disabilities

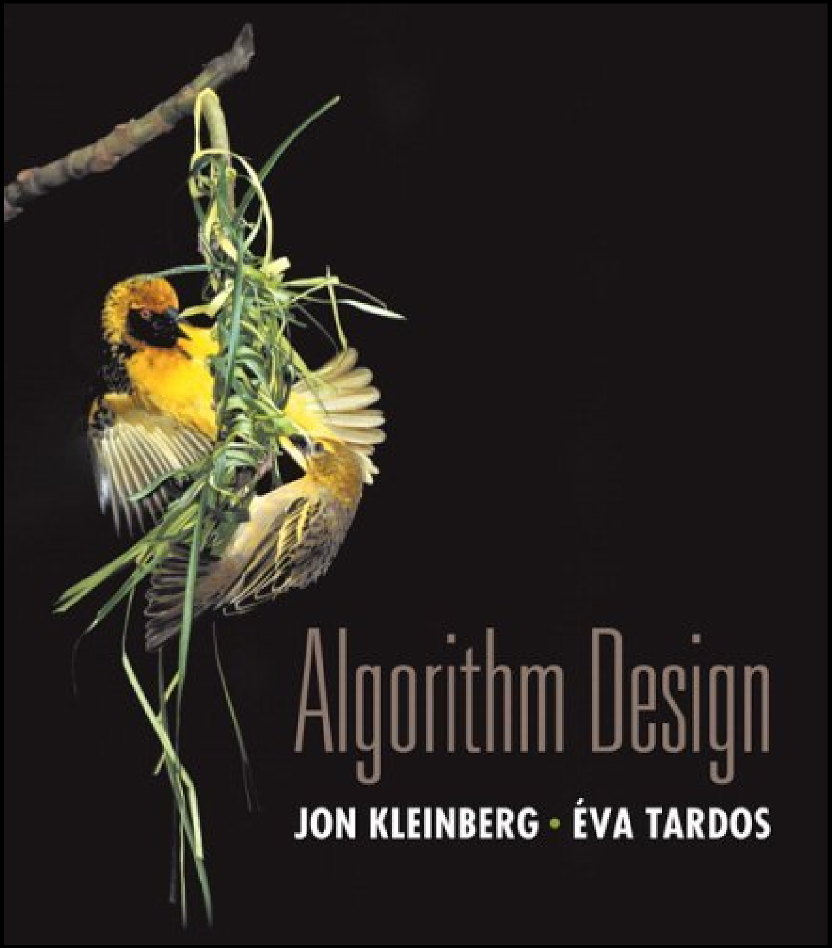
- If you have a disability that requires special academic accommodation, please make an appointment to speak with the instructor within the first three (3) weeks of the semester in order to discuss any adjustments.
- Note: We cannot arrange special accommodations without confirmation from the Disability Resource Center here at Purdue (http://www.purdue.edu/drc)

# Emergency Preparedness

- Alarm Inside → Move Outside
- Siren Outside → Move Inside (Shelter in Place)
  - Once Inside Seek Clarifying Information
    - Purdue Homepage
    - E-mail Alert
    - Purdue Emergency Warning Notification System
    - (http://www.purdue.edu/ehps/emergency_preparedness/warning-system.htm )



https://www.purdue.edu/emergency_preparedness/flipchart/index.html




# Chapter 1

## Introduction: Some Representative Problems

Slides by Kevin Wayne.

## Algorithm.

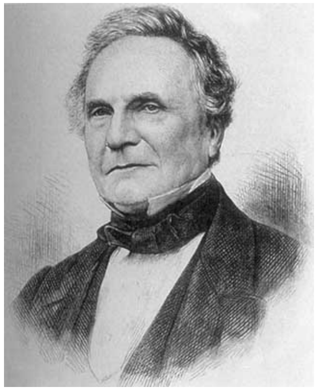
- [webster.com] A procedure for solving a mathematical problem (as of finding the greatest common divisor) in a finite number of steps that frequently involves repetition of an operation.

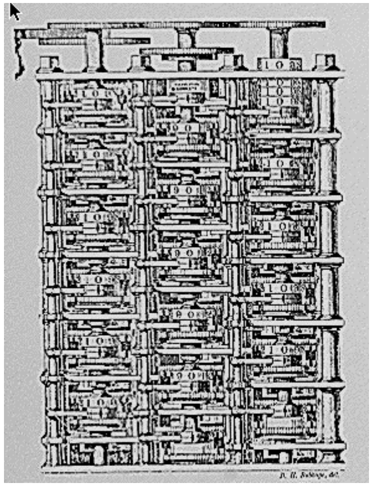
- [Knuth, TAOCP] An algorithm is a finite, definite, effective procedure, with some input and some output.

> Great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing. - *Francis Sullivan*

# Theory of Algorithms

"As soon as an Analytic Engine exists, it will necessarily guide the future course of the science. Whenever any result is sought by its aid, the question will arise - By what course of calculation can these results be arrived at by the machine in the *shortest time?* - *Charles Babbage*





# Algorithmic Paradigms

Design and analysis of computer algorithms.

- Greedy.
- Divide-and-conquer.
- Dynamic programming.
- Network flow.
- Randomized algorithms.
- Intractability.
- Coping with intractability.

Critical thinking and problem-solving.

# Applications

Wide range of applications.

- Caching.
- Compilers.
- Databases.
- Scheduling.
- Networking.
- Web Search.
- Cryptography.
- Data analysis.
- Signal processing.
- Computer graphics.
- Scientific computing.
- Operations research.
- Artificial intelligence.
- Machine Learning.
- Computational…
  - Biology
  - Physics
  - Chemistry
  - …
- National Kidney Registry.
- . . .
- National Resident Matching Program.

We focus on algorithms and techniques that are useful in practice.

# 1.1 A First Problem: Stable Matching

# Matching Problem

Goal. Given n men and n women, find a "suitable" matching.

- Participants rate members of opposite sex.
- Each man lists women in order of preference from best to worst.
- Each woman lists men in order of preference from best to worst.

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Clare | Amy | Bertha |
| Zeus | Clare | Amy | Bertha |

*Men's Preference Profile*

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Amy | Xavier | Zeus | Yancey |
| Bertha | Yancey | Xavier | Zeus |
| Clare | Yancey | Xavier | Zeus |

*Women's Preference Profile*

Candidate Assignment 1: X-A, Y-C, Z-B

Candidate Assignment 2: X-B, Y-C, Z-A

# Stable Matching Problem

Perfect matching: everyone is matched monogamously.

- Each man gets exactly one woman.
- Each woman gets exactly one man.

Unstable pair: Man X and Woman A are unstable in matching M if:

- X prefers A to his assigned partner in matching M.
- A prefers X to her assigned partner in matching M.
- Unstable pair X-A could each improve by eloping.

Stable matching: perfect matching with no unstable pairs.

- No incentive for some pair of participants to undermine assignment by joint action.

Stable matching problem. Given the preference lists of n men and n women, find a stable matching if one exists.

# Matching Residents to Hospitals

Goal. Given a set of preferences among hospitals and medical school students, design a self-reinforcing admissions process.

Unstable pair: applicant x and hospital y are unstable if:
- x prefers y to its assigned hospital.
- y prefers x to one of its admitted students.

Stable assignment. Assignment with no unstable pairs.
- Natural and desirable condition.
- Individual self-interest will prevent any applicant/hospital deal from being made.

# Stable Matching Problem

**Q.** Is assignment X-C, Y-B, Z-A stable?

favorite ↓ … least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Bertha | Amy | Clare |
| Zeus | Amy | Bertha | Clare |

*Men's Preference Profile*

favorite ↓ … least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

*Women's Preference Profile*

Note Modified Preference Lists

# Stable Matching Problem

Q. Is assignment X-C, Y-B, Z-A stable?

A. No. Bertha and Xavier will hook up.

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Bertha | Amy | Clare |
| Zeus | Amy | Bertha | Clare |

*Men's Preference Profile*

| | 1st (favorite) | 2nd | 3rd (least favorite) |
|---|---|---|---|
| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

*Women's Preference Profile*

# Stable Matching Problem

**Q.** Is assignment X-A, Y-B, Z-C stable?
**A.** Yes.

favorite ↓ … least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | Amy | Bertha | Clare |
| Yancey | Bertha | Amy | Clare |
| Zeus | Amy | Bertha | Clare |

*Men's Preference Profile*

favorite ↓ … least favorite ↓

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Yancey | Xavier | Zeus |
| Bertha | Xavier | Yancey | Zeus |
| Clare | Xavier | Yancey | Zeus |

*Women's Preference Profile*

# Stable Roommate Problem

Q. Do stable matchings always exist?
A. Not obvious a priori.

is core of market nonempty?

Stable roommate problem.

- 2n people; each person ranks others from 1 to 2n-1.
- Assign roommate pairs so that no unstable pairs.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Adam | B | C | D |
| Bob | C | A | D |
| Chris | A | B | D |
| Doofus | A | B | C |

Observation. Stable matchings do not always exist for stable roommate problem.

# Propose-And-Reject Algorithm

Propose-and-reject algorithm. [Gale-Shapley 1962] Intuitive method that guarantees to find a stable matching.

play

```
Initialize each person to be free.
while (some man is free and hasn't proposed to every woman)
{
   Choose such a man m
   w = 1st woman on m's list to whom m has not yet proposed
   if (w is free)
      assign m and w to be engaged
   else if (w prefers m to her fiancé m')
      assign m and w to be engaged, and m' to be free
   else
      w rejects m
}
```

# Proof of Correctness: Termination

Observation 1. Men propose to women in decreasing order of preference.

Observation 2. Once a woman is matched, she never becomes unmatched; she only "trades up."

Claim. Algorithm terminates after at most $n^2$ iterations of while loop.
Pf. Each time through the while loop a man proposes to a **new** woman. There are only $n^2$ possible proposals. ▪

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Victor | A | B | C | D | E |
| Wyatt | B | C | D | A | E |
| Xavier | C | D | A | B | E |
| Yancey | D | A | B | C | E |
| Zeus | A | B | C | D | E |

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Amy | W | X | Y | Z | V |
| Bertha | X | Y | Z | V | W |
| Clare | Y | Z | V | W | X |
| Diane | Z | V | W | X | Y |
| Erika | V | W | X | Y | Z |

$n(n-1) + 1$ proposals required

# Proof of Correctness: Perfection

Claim. All men and women get matched.

Pf. (by contradiction)

- Suppose, for sake of contradiction, that Zeus is not matched upon termination of algorithm.
- Then some woman, say Amy, is not matched upon termination.
- By Observation 2, Amy was never proposed to.
- But, Zeus proposes to everyone, since he ends up unmatched. ▪

# Proof of Correctness: Stability

Claim. No unstable pairs.

Pf. (by contradiction)

- Suppose A-Z is an unstable pair: each prefers each other to partner in Gale-Shapley matching S*.

- Case 1: Z never proposed to A.
  $\Rightarrow$ Z prefers his GS partner to A. (men propose in decreasing order of preference)
  $\Rightarrow$ A-Z is stable.

- Case 2: Z previously proposed to A.
  $\Rightarrow$ A rejected Z (right away or later)
  $\Rightarrow$ A prefers her GS partner to Z. $\leftarrow$ women only trade up
  $\Rightarrow$ A-Z is stable.

- In either case A-Z is stable, a contradiction. ▪

| S* |
|---|
| Amy-Yancey |
| Bertha-Zeus |
| . . . |

# Summary

Stable matching problem. Given n men and n women, and their preferences, find a stable matching if one exists.

Gale-Shapley algorithm. Guarantees to find a stable matching for any problem instance.

Q. How to implement GS algorithm efficiently?

Q. If there are multiple stable matchings, which one does GS find?

# Efficient Implementation

Efficient implementation. We describe $O(n^2)$ time implementation.

Representing men and women.

- Assume men are named 1, …, n.
- Assume women are named 1', …, n'.

Engagements.

- Maintain a list of free men, e.g., in a queue.
- Maintain two arrays `wife[m]`, and `husband[w]`.
  - set entry to `0` if unmatched
  - if m matched to w then `wife[m]=w` and `husband[w]=m`

Men proposing.

- For each man, maintain a list of women, ordered by preference.
- Maintain an array `count[m]` that counts the number of proposals made by man `m`.

# Efficient Implementation

Women rejecting/accepting.

- Does woman `w` prefer man `m` to man `m'`?
- For each woman, create inverse of preference list of men.
- Constant time access for each query after O(n) preprocessing.

| Amy | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th |
|---|---|---|---|---|---|---|---|---|
| Pref | 8 | 3 | 7 | 1 | 4 | 5 | 6 | 2 |

| Amy | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Inverse | 4th | 8th | 2nd | 5th | 6th | 7th | 3rd | 1st |

```
for i = 1 to n
   inverse[pref[i]] = i
```

Amy prefers man 3 to 6
since `inverse[3]` < `inverse[6]`

2 7

## Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

An instance with two stable matchings.

- A-X, B-Y, C-Z.
- A-Y, B-X, C-Z.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | A | B | C |
| Yancey | B | A | C |
| Zeus | A | B | C |

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Y | X | Z |
| Bertha | X | Y | Z |
| Clare | X | Y | Z |

## Understanding the Solution

Q. For a given problem instance, there may be several stable matchings. Do all executions of Gale-Shapley yield the same stable matching? If so, which one?

Def. Man m is a valid partner of woman w if there exists some stable matching in which they are matched.

Man-optimal assignment. Each man receives best valid partner.

Claim. All executions of GS yield man-optimal assignment, which is a stable matching!

- No reason a priori to believe that man-optimal assignment is perfect, let alone stable.
- Simultaneously best for each and every man.

# Man Optimality

Claim. GS matching S* is man-optimal.

Pf. (by contradiction)

- Suppose some man is paired with someone other than best partner. Men propose in decreasing order of preference $\Rightarrow$ some man is rejected by valid partner.
- Let Y be first such man, and let A be first valid woman that rejects him.
- Let S be a stable matching where A and Y are matched.
- When Y is rejected, A forms (or reaffirms) engagement with a man, say Z, whom she prefers to Y.
- Let B be Z's partner in S.
- Z not rejected by any valid partner at the point when Y is rejected by A. Thus, Z prefers A to B.
- But A prefers Z to Y.
- Thus A-Z is unstable in S. ▪

since this is first rejection by a valid partner

| S |
|---|
| Amy-Yancey |
| Bertha-Zeus |
| . . . |

# Stable Matching Summary

Stable matching problem. Given preference profiles of n men and n women, find a stable matching.

no man and woman prefer to be with each other than assigned partner

Gale-Shapley algorithm. Finds a stable matching in $O(n^2)$ time.

Man-optimality. In version of GS where men propose, each man receives best valid partner.

w is a valid partner of m if there exist some stable matching where m and w are paired

Q. Does man-optimality come at the expense of the women?

# Woman Pessimality

Woman-pessimal assignment. Each woman receives worst valid partner.

Claim. GS finds woman-pessimal stable matching S*.

Pf.

- Suppose A-Z matched in S*, but Z is not worst valid partner for A.
- There exists stable matching S in which A is paired with a man, say Y, whom she likes less than Z.
- Let B be Z's partner in S.
- Z prefers A to B. ← man-optimality
- Thus, A-Z is an unstable in S. ▪

| S |
|---|
| Amy-Yancey |
| Bertha-Zeus |
| . . . |

# Extensions: Matching Residents to Hospitals

Ex: Men ≈ hospitals, Women ≈ med school residents.

Variant 1. Some participants declare others as unacceptable.

resident A unwilling to work in Cleveland

Variant 2. Unequal number of men and women.

Variant 3. Limited polygamy.

hospital X wants to hire 3 residents

Def. Matching S unstable if there is a hospital h and resident r such that:

- h and r are acceptable to each other; and
- either r is unmatched, or r prefers h to her assigned hospital; and
- either h does not have all its places filled, or h prefers r to *at least one* of its assigned residents.

# Application: Matching Residents to Hospitals

NRMP. (National Resident Matching Program)

- Original use just after WWII. ← predates computer usage
- Ides of March, 23,000+ residents.

Rural hospital dilemma.

- Certain hospitals (mainly in rural areas) were unpopular and declared unacceptable by many residents.
- Rural hospitals were under-subscribed in NRMP matching.
- How can we find stable matching that benefits "rural hospitals"?

Rural Hospital Theorem. Rural hospitals get exactly same residents in every stable matching!

# Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Potentially deep social ramifications. [legal disclaimer]

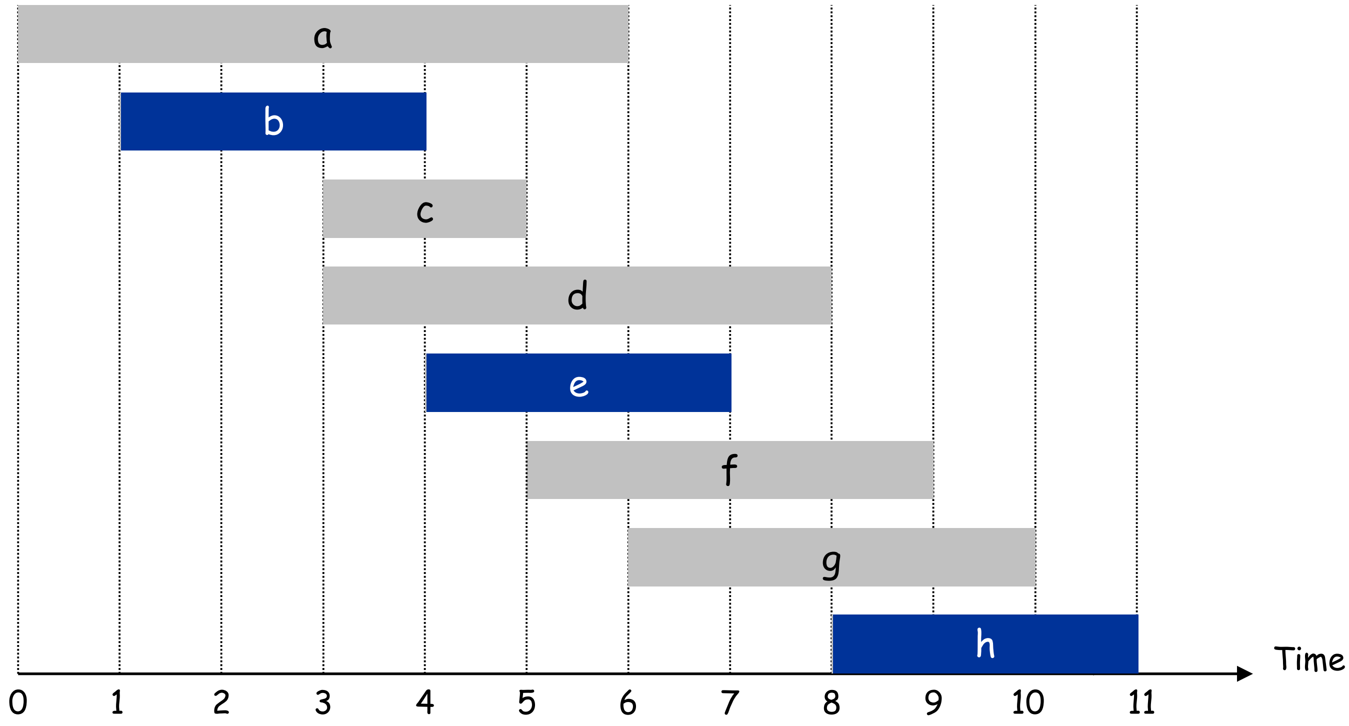
# 1.2 Five Representative Problems

# Interval Scheduling

Input. Set of jobs with start times and finish times.

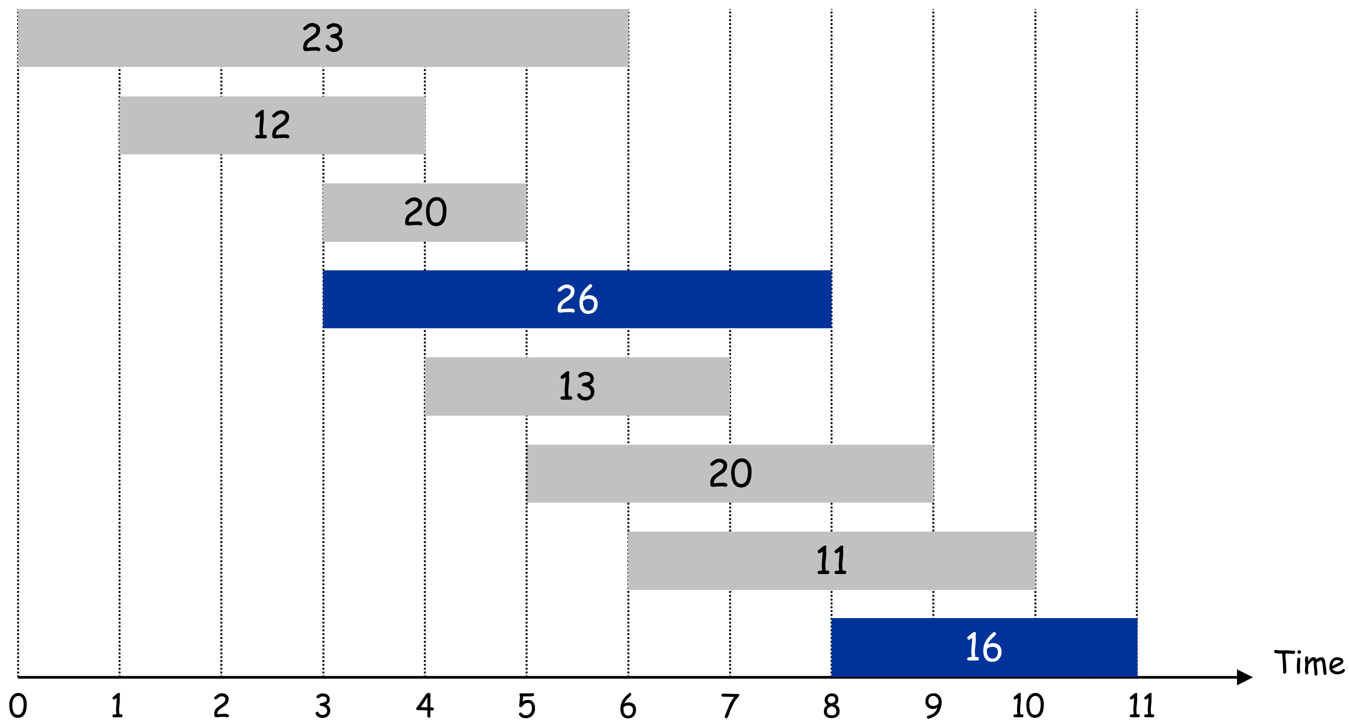
Goal. Find maximum cardinality subset of mutually compatible jobs.

jobs don't overlap



# Weighted Interval Scheduling

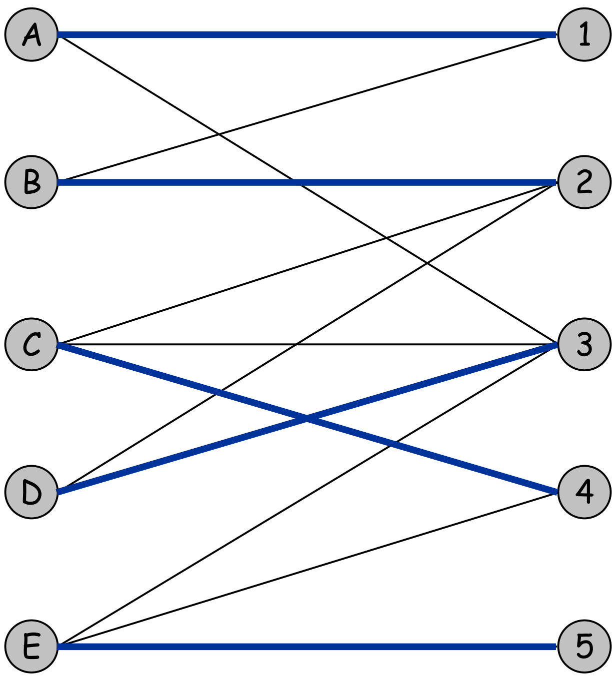
Input. Set of jobs with start times, finish times, and weights.
Goal. Find maximum weight subset of mutually compatible jobs.



# Bipartite Matching

Input. Bipartite graph.

Goal. Find maximum cardinality matching.

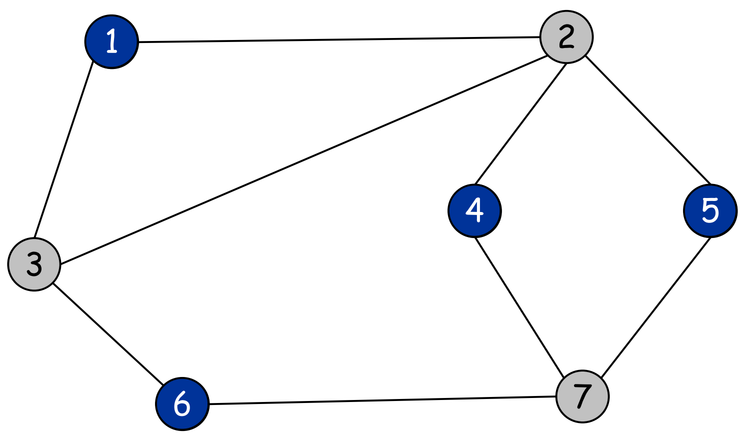


# Independent Set

Input. Graph.

Goal. Find maximum cardinality independent set.

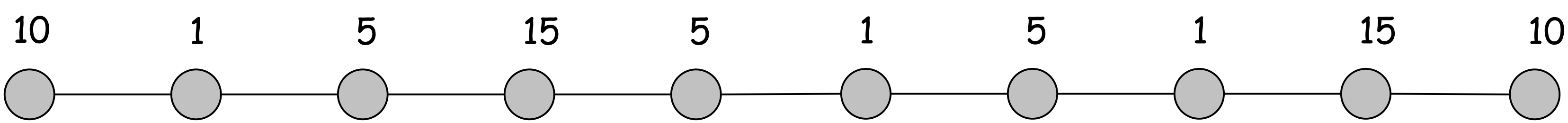
subset of nodes such that no two joined by an edge



# Competitive Facility Location

**Input.** Graph with weight on each node.
**Game.** Two competing players alternate in selecting nodes.
Not allowed to select a node if any of its neighbors have been selected.

**Goal.** Select a maximum weight subset of nodes.

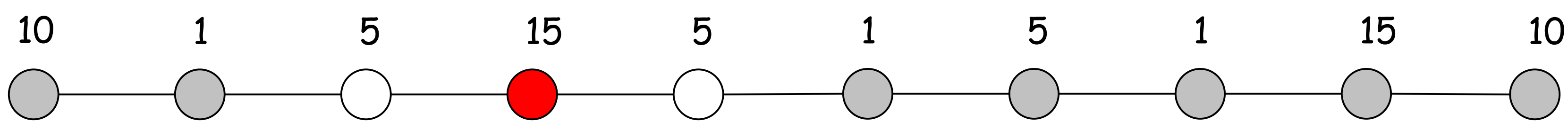



Second player can guarantee 20, but not 25.

# Competitive Facility Location

Input. Graph with weight on each node.
Game. Two competing players alternate in selecting nodes.
Not allowed to select a node if any of its neighbors have been selected.

Goal. Select a maximum weight subset of nodes.

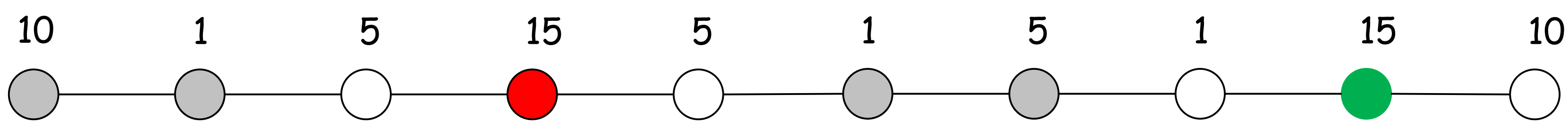



Second player can guarantee 20, but not 25.

# Competitive Facility Location

**Input.** Graph with weight on each node.
**Game.** Two competing players alternate in selecting nodes.
Not allowed to select a node if any of its neighbors have been selected.

**Goal.** Select a maximum weight subset of nodes.

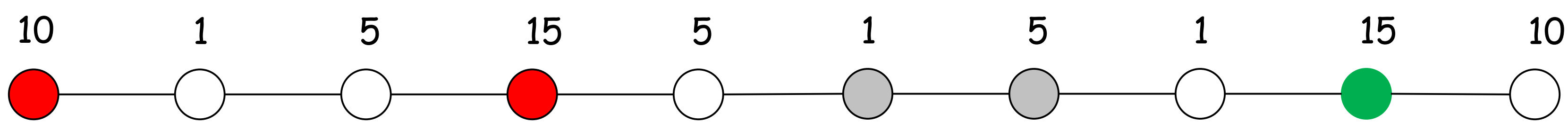



Second player can guarantee 20, but not 25.

# Competitive Facility Location

Input. Graph with weight on each node.
Game. Two competing players alternate in selecting nodes.
Not allowed to select a node if any of its neighbors have been selected.

Goal. Select a maximum weight subset of nodes.

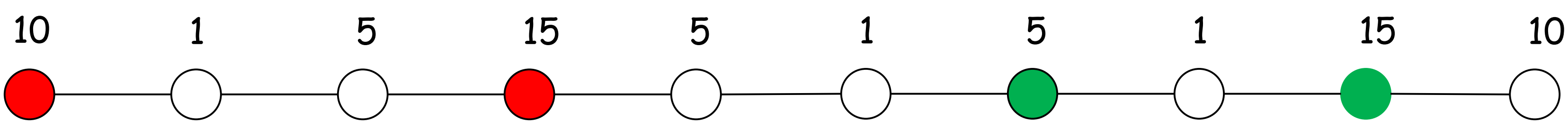



Second player can guarantee 20, but not 25.

# Competitive Facility Location

Input. Graph with weight on each node.
Game. Two competing players alternate in selecting nodes.
Not allowed to select a node if any of its neighbors have been selected.

Goal. Select a maximum weight subset of nodes.




Second player can guarantee 20, but not 25.

# Five Representative Problems

Variations on a theme: independent set.

Interval scheduling: n log n greedy algorithm.
Weighted interval scheduling: n log n dynamic programming algorithm.
Bipartite matching: $n^k$ max-flow based algorithm.
Independent set: NP-complete.
Competitive facility location: PSPACE-complete.

# Extra Slides

# Stable Matching Problem

Goal: Given n men and n women, find a "suitable" matching.

- Participants rate members of opposite sex.
- Each man lists women in order of preference from best to worst.
- Each woman lists men in order of preference from best to worst.

favorite ↓ (1st) … least favorite ↓ (5th)

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Victor | Bertha | Amy | Diane | Erika | Clare |
| Wyatt | Diane | Bertha | Amy | Clare | Erika |
| Xavier | Bertha | Erika | Clare | Diane | Amy |
| Yancey | Amy | Diane | Clare | Bertha | Erika |
| Zeus | Bertha | Diane | Amy | Erika | Clare |

*Men's Preference List*

# Stable Matching Problem

Goal: Given n men and n women, find a "suitable" matching.

- Participants rate members of opposite sex.
- Each man lists women in order of preference from best to worst.
- Each woman lists men in order of preference from best to worst.

favorite ↓ (1st) … least favorite ↓ (5th)

| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Amy | Zeus | Victor | Wyatt | Yancey | Xavier |
| Bertha | Xavier | Wyatt | Yancey | Victor | Zeus |
| Clare | Wyatt | Xavier | Yancey | Zeus | Victor |
| Diane | Victor | Zeus | Yancey | Xavier | Wyatt |
| Erika | Yancey | Wyatt | Zeus | Xavier | Victor |

*Women's Preference List*

# Understanding the Solution

Claim. The man-optimal stable matching is weakly Pareto optimal.

No other perfect matching (stable or unstable) where every man does strictly better

Pf.

- Let A be last woman in some execution of GS algorithm to receive a proposal.
- No man is rejected by A since algorithm terminates when last woman receives first proposal.
- No man matched to A will be strictly better off than in man-optimal stable matching. ▪

# Deceit: Machiavelli Meets Gale-Shapley

Q. Can there be an incentive to misrepresent your preference profile?

- Assume you know men's propose-and-reject algorithm will be run.
- Assume that you know the preference profiles of all other participants.

Fact. No, for any man yes, for some women. No mechanism can guarantee a stable matching and be cheatproof.

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Xavier | A | B | C |
| Yancey | B | A | C |
| Zeus | A | B | C |

*Men's Preference List*

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Y | X | Z |
| Bertha | X | Y | Z |
| Clare | X | Y | Z |

*Women's True Preference Profile*

| | 1st | 2nd | 3rd |
|---|---|---|---|
| Amy | Y | Z | X |
| Bertha | X | Y | Z |
| Clare | X | Y | Z |

*Amy Lies*

# Lessons Learned

Powerful ideas learned in course.

- Isolate underlying structure of problem.
- Create useful and efficient algorithms.

Potentially deep social ramifications. [legal disclaimer]

- Historically, men propose to women. Why not vice versa?
- Men: propose early and often.
- Men: be more honest.
- Women: ask out the guys.
- Theory can be socially enriching and fun!
- CS majors get the best partners!