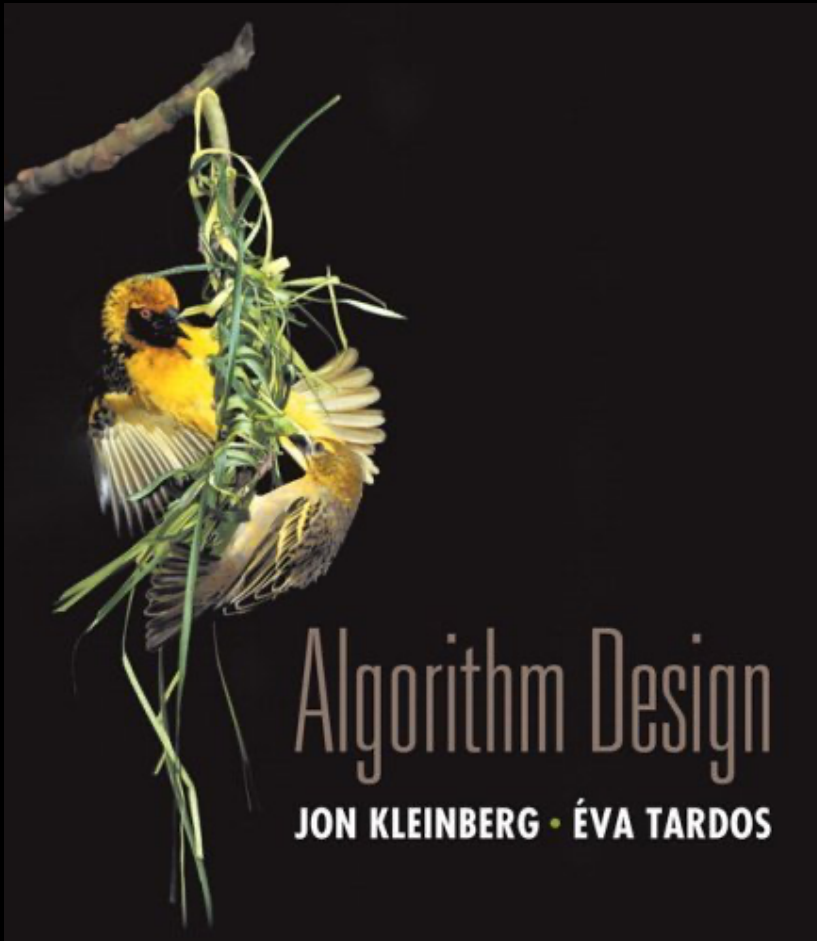


Chapter 1

Introduction: Some Representative Problems



Slides by Kevin Wayne.
Copyright © 2005 Pearson-Addison Wesley.
All rights reserved.
Updated by T. K. Prasad

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.

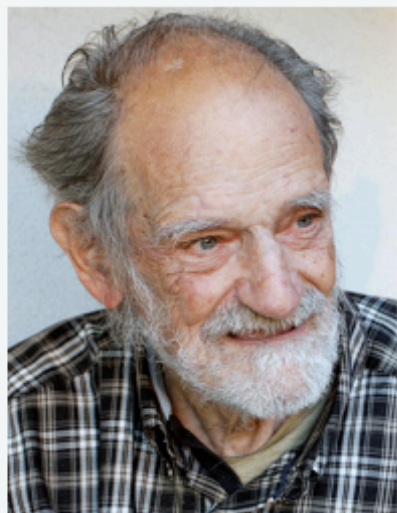
Stable assignment. Assignment with no unstable pairs.

- Natural and desirable condition.
- Individual self-interest will prevent any new applicant/hospital deal from being made or broken.

2012 Nobel Prize in Economics

Lloyd Shapley. Stable matching theory and Gale-Shapley algorithm. (1962)

Alvin Roth. Applied Gale-Shapley to matching new doctors with hospitals, students with schools, and organ donors with patients.



Lloyd Shapley



Alvin Roth



Stable Marriage/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 ↓ 1 st		least favorite ↓ 3 rd
	1 st	2 nd	3 rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite ↓ 1 st		least favorite ↓ 3 rd
	1 st	2 nd	3 rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

Perfect matching: everyone is matched monogamously.

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

Stability: no incentive for some pair of participants to undermine assignment by joint action.

- In matching M , an unmatched pair m - w is **unstable** if man m and woman w prefer each other to current partners.
- Unstable pair m - w could each improve by eloping.

Stable matching: perfect matching with no unstable pairs.

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

Stable Matching Problem

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

	favorite ↓ 1 st	2 nd	least favorite ↓ 3 rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite ↓ 1 st	2 nd	least favorite ↓ 3 rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Problem

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

A. No. Bertha and Xavier will hook up.

	favorite ↓ 1 st	2 nd	least favorite ↓ 3 rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite ↓ 1 st	2 nd	least favorite ↓ 3 rd
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 ↓ 1 st		least favorite ↓ 3 rd
	1 st	2 nd	3 rd
Xavier	Amy	Bertha	Clare
Yancey	Bertha	Amy	Clare
Zeus	Amy	Bertha	Clare

Men's Preference Profile

	favorite ↓ 1 st		least favorite ↓ 3 rd
	1 st	2 nd	3 rd
Amy	Yancey	Xavier	Zeus
Bertha	Xavier	Yancey	Zeus
Clare	Xavier	Yancey	Zeus

Women's Preference Profile

Stable Matching Examples

M prefers W over W'
M' prefers W over W'
W prefers M over M'
W' prefers M' over M

Unique Stable Match: M-W M'-W'

M prefers W' over W
M' prefers W over W'
W prefers M over M'
W' prefers M' over M

One Stable Match: M-W' M'-W

Another Stable Match: M-W M'-W'

(Former is M-biased while latter is W-biased.)

Stable Roommate Problem

Q. Do stable matchings always exist?

A. Not obvious a priori.

Stable roommate problem.

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

	<i>1st</i>	<i>2nd</i>	<i>3rd</i>
<i>Adam</i>	B	C	D
<i>Bob</i>	C	A	D
<i>Chris</i>	A	B	D
<i>Doofus</i>	A	B	C

$A-B, C-D \Rightarrow B-C$ unstable
 $A-C, B-D \Rightarrow A-B$ unstable
 $A-D, B-C \Rightarrow A-C$ unstable

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

Theoretical Questions

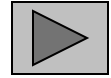
Does there exist a stable matching for every set of preference lists?

Given a set of preference lists, can we efficiently construct a stable matching?

If there are multiple stable matching, which one does the algorithm construct? Is it the same on each run (given the nondeterminism)?

Propose-And-Reject Algorithm

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



```
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. •

	1 st	2 nd	3 rd	4 th	5 th
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

	1 st	2 nd	3 rd	4 th	5 th
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 - realizing the worst case bound

Proof of Correctness: Perfection

Claim. All men and women get matched.

Pf. (by contradiction)

- Suppose, for the 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 has proposed to everyone, since he ends up unmatched at termination. ▪

[contradiction]

Proof of Correctness: Stability

Claim. No unstable pairs.

Pf. (by contradiction)

- Suppose $A-Z$ causes instability: each prefers the other to partner in Gale-Shapley matching S^* .

- **Case 1: Z never proposed to A .**
 - $\Rightarrow Z$ prefers his GS partner to A .
 - $\Rightarrow A-Z$ does not cause instability.

men propose in decreasing
order of preference

S^*

Amy-Yancey

Bertha-Zeus

...

- **Case 2: Z proposed to A .**
 - $\Rightarrow A$ rejected Z (right away or later)
 - $\Rightarrow A$ prefers her GS partner to Z .
 - $\Rightarrow A-Z$ does not cause instability.

women only trade up

- In either case $A-Z$ assignment is not “stable”, a contradiction.

Summary

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

Gale-Shapley algorithm. Guaranteed 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

Termination argument shows that the algorithm requires at most n^2 iterations.

To show that the algorithm takes time (and storage) proportional to n^2 ($O(n^2)$), we design data structures such that each iteration takes a bounded number of steps, each step requiring constant time (independent of the input size) to execute.

Note that because the input size is $O(n^2)$, the algorithm is technically linear.

Efficient Implementation

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

Representing men and women.

- Assume men are named $1, \dots, n$.
- Assume women are named $1', \dots, n'$.

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 an array 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	1 st	2 nd	3 rd	4 th	5 th	6 th	7 th	8 th
Pref	8	3	7	1	4	5	6	2

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

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

Amy prefers man 3 to 6
since $\text{inverse}[3] < \text{inverse}[6]$
2 7

Efficient Implementation

Verify that each step below takes constant time:

- (1) Free man can be picked from **the list**, say the first element.
- (2) Highest-ranked woman not-yet-proposed can be determined using **count** and **man-pref** arrays.
- (3) Woman's freedom is determined using **husband**-array.
- (4) Engagement requires two assignments - one to **husband** array and another to **wife** array.
- (5) Woman can compare options using **inverse** array (derived by preprocessing **woman-pref** array).

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.

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

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

Understanding the Solution: Uniqueness of GS Matching

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

Examples

- Both **Amy** and **Bertha** are **valid partners** for Xavier.
- Both **Bertha** and **Amy** are **valid partners** for Yancey.
- Clare is the only **valid partner** for Zeus.

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

Male-optimal (Male-biased) assignment:

Xavier-**Amy**, Yancey-**Bertha**, Zeus-Clare

Female-optimal (Female-biased) assignment:

Amy-Yancey, **Bertha**-Xavier, Clare-Zeus

Uniqueness of GS Matching

Claim. All executions of GS yield **man-optimal** assignment. And it is a stable matching!

Three Surprising Observations:

- No reason to believe that man-optimal assignment is *perfect*, let alone *stable*.
- **Simultaneously best** for each and every man.
- In spite of the non-determinism associated with the selection of the man who proposes, the man-optimal assignment is fixed.

Man Optimality

Claim. GS matching S^* is man-optimal (and hence unique).

Pf. (by contradiction)

- In a GS matching S^* :
 - let m be the **first** to be rejected by $w = \text{best}(m)$
as w prefers some m' over m [i]
- In another **stable** matching S :
 - let m be matched with $w = \text{best}(m)$
and m' be paired with some w'
- In matching S^* : consider m'
 - since m rejection is the **first**
 m' already proposed to w over w' [ii]
(This is why m' is hooked up with w
and m' was not rejected by valid partner w' earlier.)
- w prefers m' over m from [i]
- m' prefers w over w' from [ii]
- S has instability due to w and m' $\Rightarrow$ **Contradiction**

Stable Matching Summary

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

no unassigned man-woman pair prefer to be with each other than their assigned partners

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

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

w is a valid partner of m if there exists 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 matching S^* .

Pf. (by contradiction)

- Let (m, w) in S^* is such that m is not worst(w). Also, from man-optimal result, $w = \text{best}(m)$. [i]
- Then there exists another stable matching S in which w is paired with $m' = \text{worst}(w)$.
 - ✎ w prefers m to m' [ii]
- In S , m is paired with some $w' \neq w$.
 - ✎ m prefers w to w' [iii]
because $w = \text{best}(m)$ (from [i]).
- Thus, m - w causes instability in S . $\Rightarrow$ contradiction.

Overall Takeaway

Problem Instance: Practical problem of significance

Proof of correctness of algorithm pseudo code: There exists a stable solution *for any preference lists*

Algorithm Analysis and Data Structure Design

Best valid partner (global) definition

Proof that GS Alg. computes Men-Optimal / Women-Pessimal solution:

Nondeterminism immaterial - Unique solution

Best valid partner assignment: perfect and stable

Extensions: Matching Residents to Hospitals

Ex: Men $\approx$ hospitals, Women $\approx$ med school residents.

Variant 1. Some participants declare others as unacceptable.

Variant 2. Unequal number of men and women.

resident A unwilling to
work in Cleveland

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
- Ends 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!

Powerful ideas learned in course.

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

1.2 Five Representative Problems

All problems related at an abstract-level,
but subtly different at the next level.

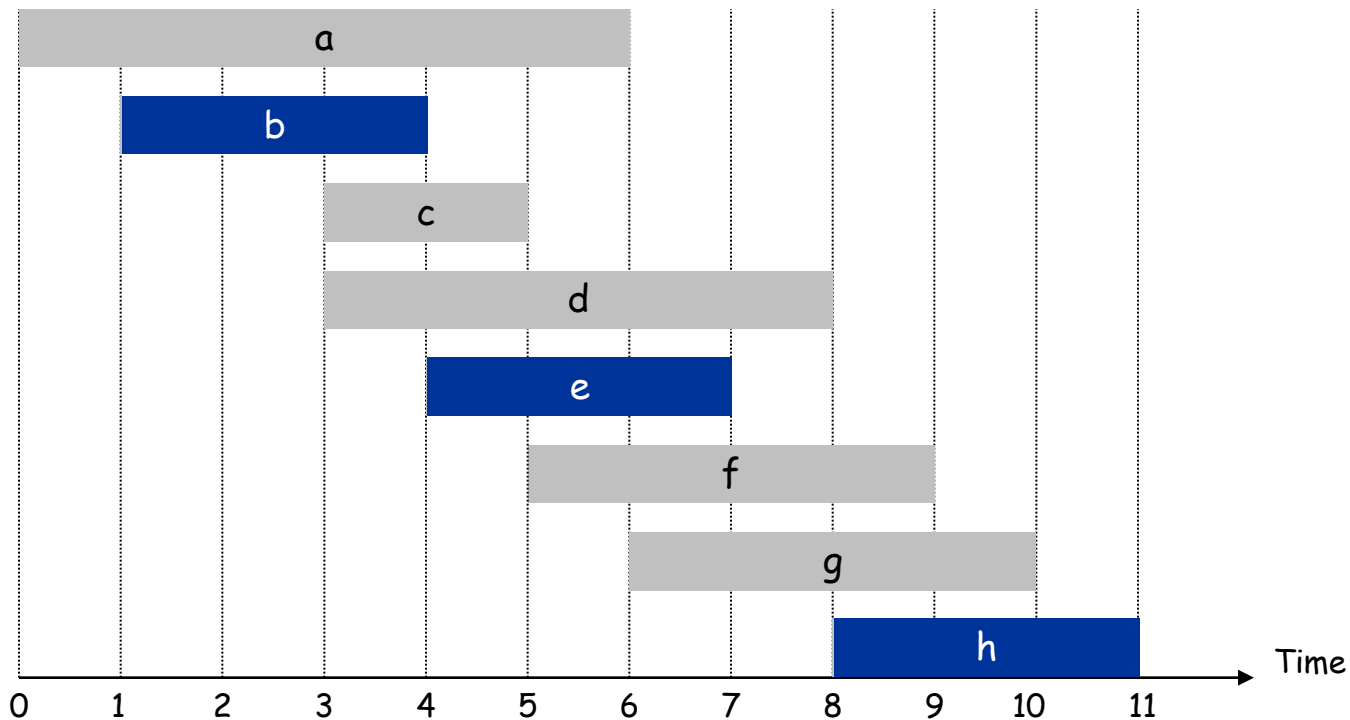
Require strategies of significantly
different kinds and algorithms of
different complexities.

Interval Scheduling

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

Goal. Find **maximum cardinality** subset of mutually compatible jobs.

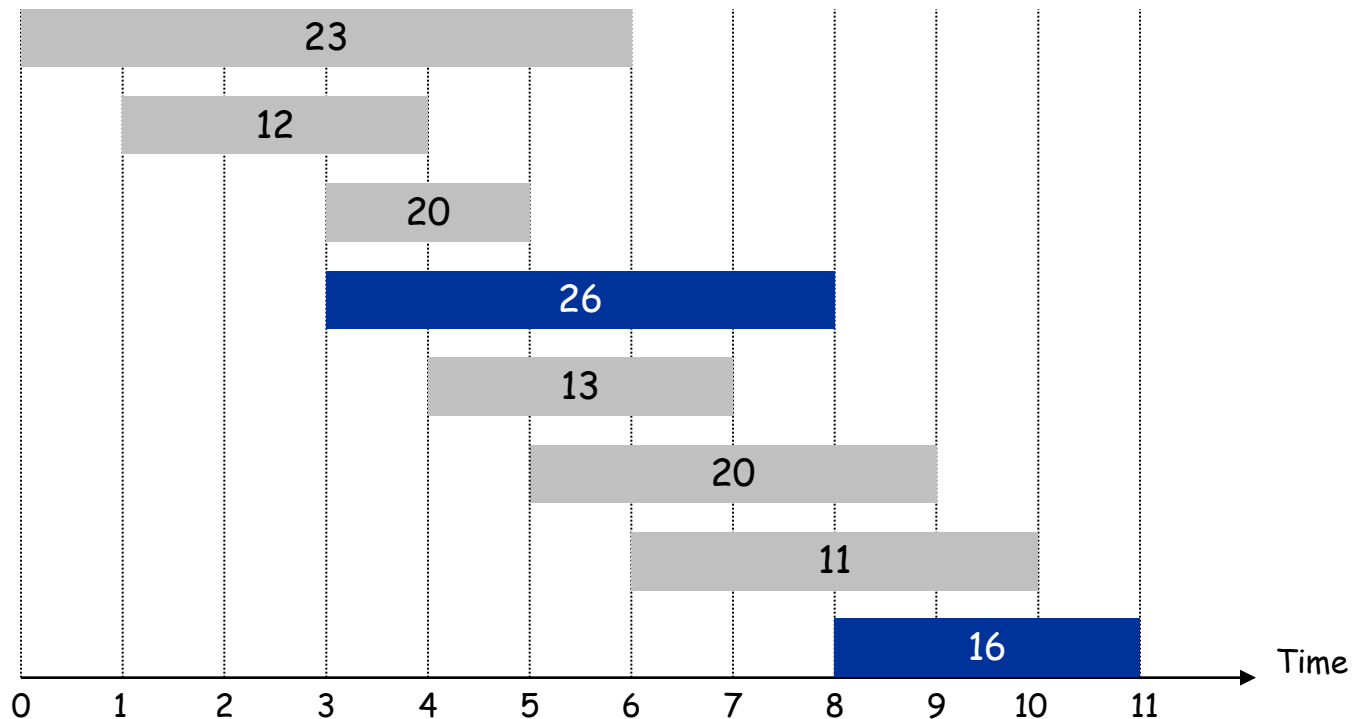
↑
jobs don't overlap



Weighted Interval Scheduling

Input. Set of jobs with start times, finish times, and **weights**.

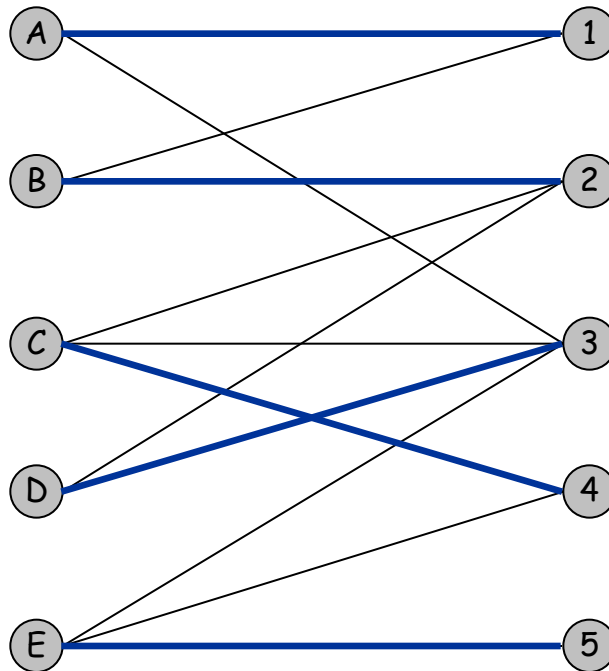
Goal. Find **maximum weight** subset of mutually compatible jobs.



Bipartite Matching

Input. Bipartite graph.

Goal. Find **maximum cardinality** matching.

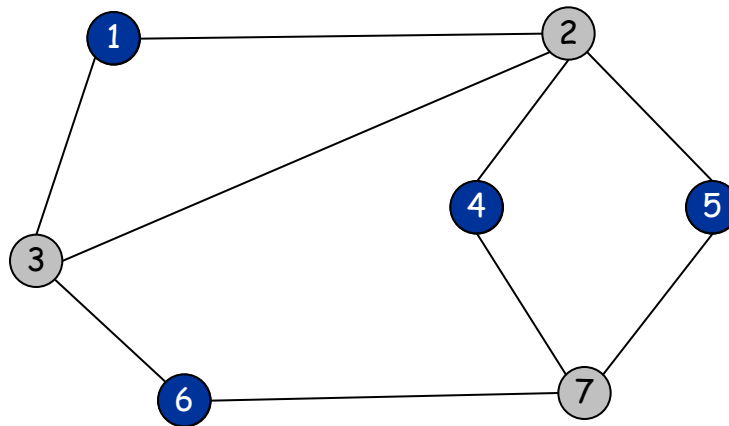


Independent Set

Input. Graph.

Goal. Find **maximum cardinality independent set**.

↑
subset of nodes such that no two nodes
are joined by an edge

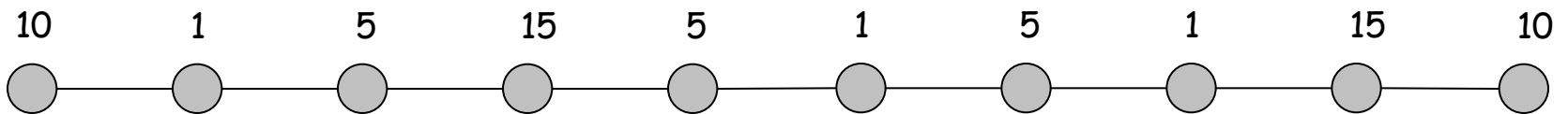


Competitive Facility Location

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.

Five Representative Problems

Variations on a theme:

"maximal 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.