Greedy Algorithms

- Stable Matching

Stable Matching

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

Men's Preference Profile

	favorite ↓ 1st	2nd	least favorite ↓ 3rd
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.

	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.

	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 Roommate Problem

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

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

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 matching 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.

```
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 proposed to A.
  $\Rightarrow$ A rejected Z (right away or later)
  $\Rightarrow$ A prefers her GS partner to Z. ← 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.

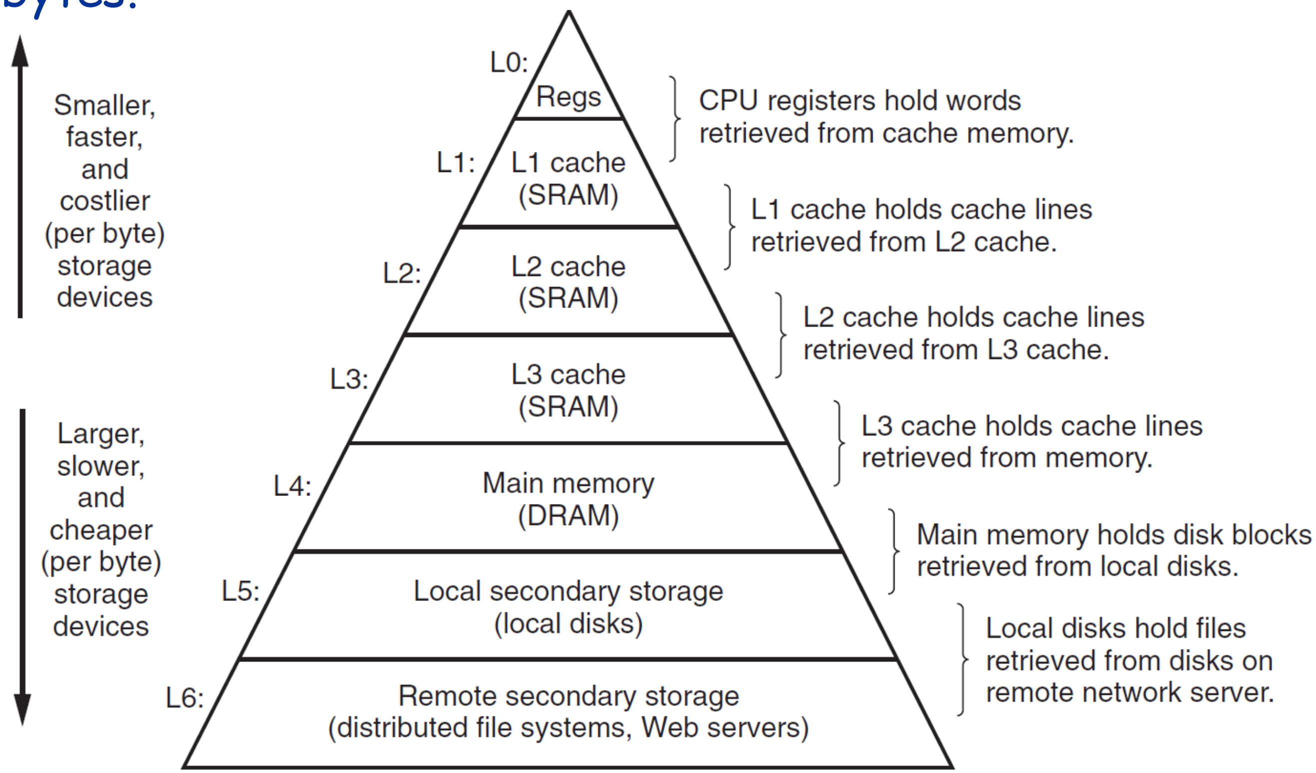
- Suppose A-Z matched in S*, but Z is not worst valid partner for A.
- There exists stable matching S in which A is paired with a man, say Y, whom she likes less than Z.
- Let B be Z's partner in S.
- Z prefers A to B. ← man-optimality
- Thus, A-Z is an unstable in S. ▪

S
Amy-Yancey
Bertha-Zeus
. . .

Offline Caching

Caching

- In a computer, a cache is memory that is smaller but faster than main memory.
- It holds a small subset of what's in main memory.
- Caches store data in blocks, also known as cache lines, usually 32, 64, or 128 bytes.



Access Cache

- A program makes a sequence of memory requests to blocks. Each block usually has several requests to some data that it holds.
- The cache size is limited to k blocks, starting out empty before the first request.
- Each request causes either 0 or 1 block to enter the cache, and either 0 or 1 block
- to be evicted. A request for block b may have one of three outcomes:
  - Cache Hit
  - Cache Miss (Empty block exists)
  - Cache Miss (One block must be evicted)
- **Goal**: Given a sequence of block requests, minimize the number of cache misses.

Offline Caching

- Application Examples:
  - Sometimes you do know the entire request sequence in advance. E.g. Disk R/W planner algorithms
  - Use for baseline for online algorithms
  - Have scheduled events and you need agents to serve them.

1, 5, 5, 4, 2, 6, 3, 2, 1, 5, 4

K = 4

- LRU
- FIFO
- FIF (LFD) : Strategy for offline caching: evict the block whose next access in the request sequence comes furthest in the future.
- Intuitively, makes sense—don't keep the block if you're not going to need it soon.

Proof

- Proof idea: For any other algorithm ALG, the cost of ALG is not increased if in the 1st time that A differs from FIF we evict in A the page that is requested farthest in the future.
- Proof:
  - Consider any ALG, and define ALG_i, which has the identical steps to ALG until the i^{th} request.
  - Claim: Choosing FIF dose not increase the cost.
  - Consider ALG's cache after the i^{th} choice is X + v and ALG_i = X + u, where X is the k – 1 common pages and u and v are the difference.
  - Afterward, ALG_i imitates ALG's behavior.
    - If ALG evict v, ALG_i evicts u and the proof concluded.
    - If v requested eventually, ALG don't incur a page fault despite ALG_i. However before that u was requested.
  - Hence, the number page faults doesn't increase.
- Applying i = 1 to n to any algorithm (include OPT) results FIF.

References

References

- Section 1.1 of the text book “algorithm design” by Jon Kleinberg and Eva Tardos
- The original slides were prepared by Kevin Wayne. The slides are distributed by Pearson Addison-Wesley.