

Lecture 10: Greedy Algorithms 2

CS 341: Algorithms

Thursday, Feb 7th 2019

Example (1)

| Companies | Interns | |
|-----------|---------|---|
| 1 | A | B |
| 2 | A | B |



| Interns | Companies | |
|---------|-----------|---|
| A | 1 | 2 |
| B | 1 | 2 |

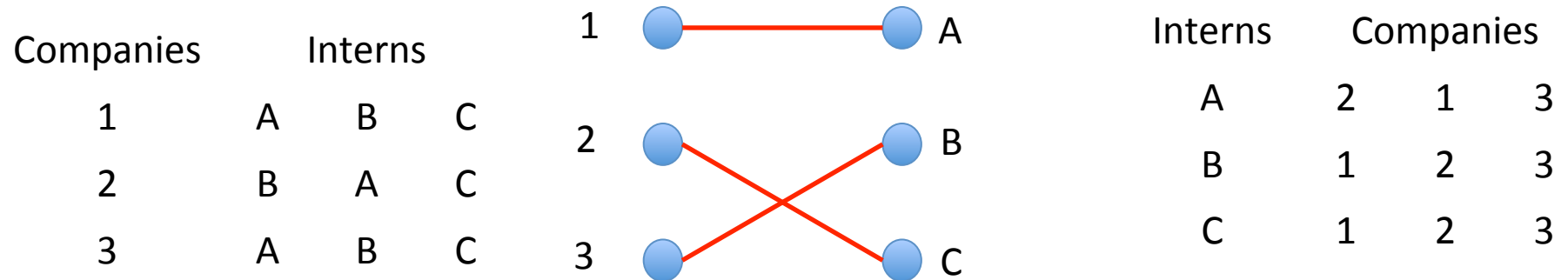
Observation: 1 and A mutually prefer each other the most.

So, in any matching they have to be matched.

Q: Is this matching stable?

A: Yes

Example (2)

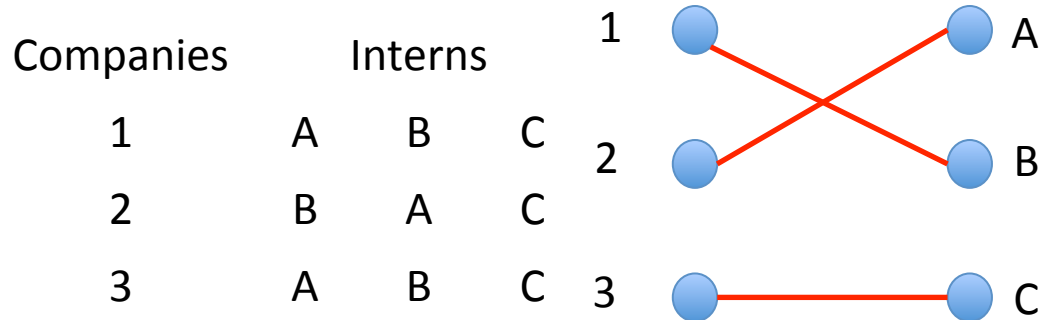


Q: Is this matching stable?

A: No

B and 2 would rather match together.

Example (2)



| Interns | Companies | | |
|---------|-----------|---|---|
| A | 2 | 1 | 3 |
| B | 1 | 2 | 3 |
| C | 1 | 2 | 3 |

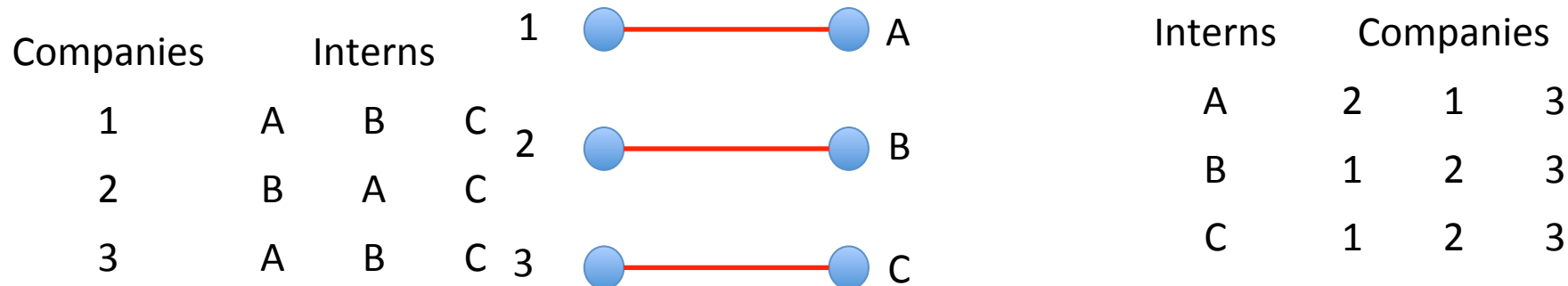
Q: Is this matching stable?

A: Yes

(Both A & B are matched with their top picks, so they wouldn't switch.

C might want to switch to 1 or 2 but C is ranked lowest for 1 and 2)

Example (2)



Q: Is this matching stable?

A: Yes

1 and 2 are matched to their top interns, so they wouldn't switch.

3 would switch to A and B but 3 is ranked lowest by A and B

Questions

1. Does a stable matching always exist *for any preference list*?

◆ How can we know that for some preference lists,
companies&interns will not keep oscillating?

2. If a stable matching always exists, can we find it efficiently?

3. If a stable matching always exists, is it unique?

Answer: No (we found 2 stable matchings in the previous example)

Gale & Shapley's Greedy Algorithm (1962)



David Gale



Lloyd Shapley

Shapley was a 2012 Nobel prize winner for his research into matching markets.

Stable Matching has many applications: residency-hospital matching, kidney-donor matching, etc.

Gale & Shapley's Greedy Algorithm (1962)

Companies

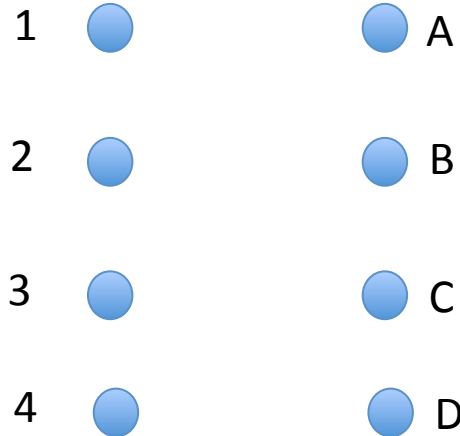
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



Initially every company & intern is unmatched
Pick any unmatched company.

Gale & Shapley's Greedy Algorithm (1962)

Companies

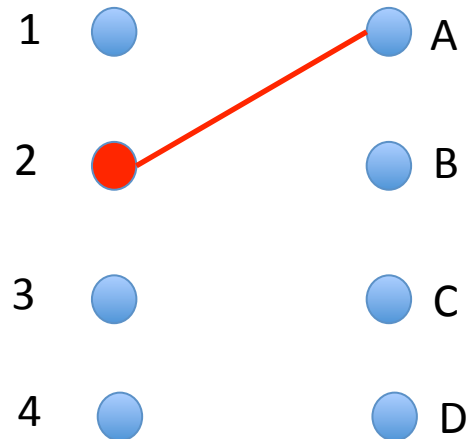
Interns

Interns

Companies

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



Picked company proposes to highest ranked intern not yet proposed to.

2 Proposes to A.

A is free; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

Companies

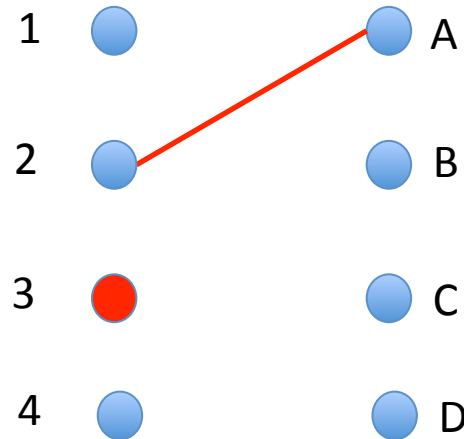
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



3 proposes to A.

A is not free; but prefers 3 to 2; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

Companies

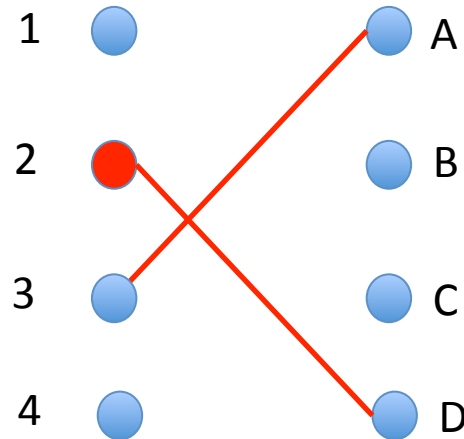
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



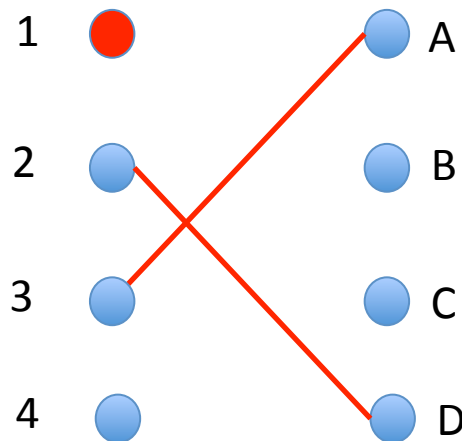
2 proposes to D (b/c already proposed to A).

D is free; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

| Companies | Interns | | | |
|-----------|---------|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

| Interns | | | | Companies |
|---------|---|---|---|-----------|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



1 proposes to A.

A prefers 1 to 3; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

Companies

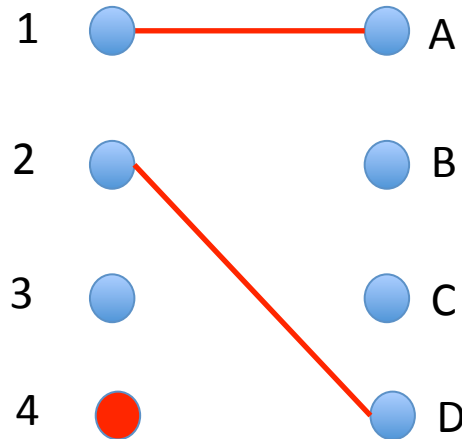
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



4 proposes to A.

A prefers 1 to 4; so rejects.

Gale & Shapley's Greedy Algorithm (1962)

Companies

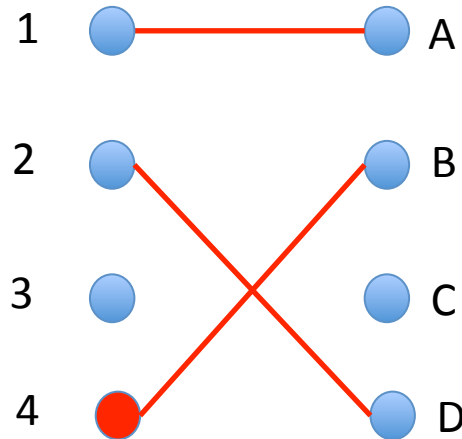
Interns

| | | | | |
|---|---|----------|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



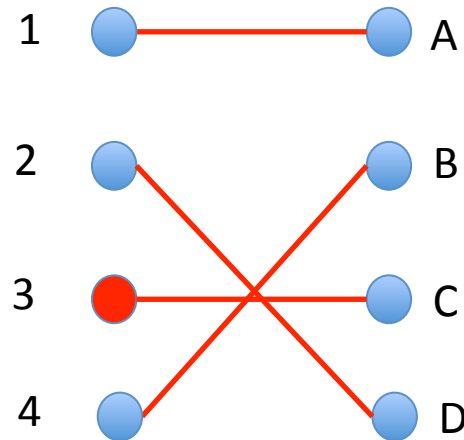
4 proposes to B.

B is free; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

| Companies | Interns | | | |
|-----------|---------|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

| Interns | | Companies | | | |
|---------|---|-----------|---|---|--|
| A | 1 | 3 | 2 | 4 | |
| B | 4 | 3 | 2 | 1 | |
| C | 2 | 3 | 1 | 4 | |
| D | 3 | 4 | 2 | 1 | |

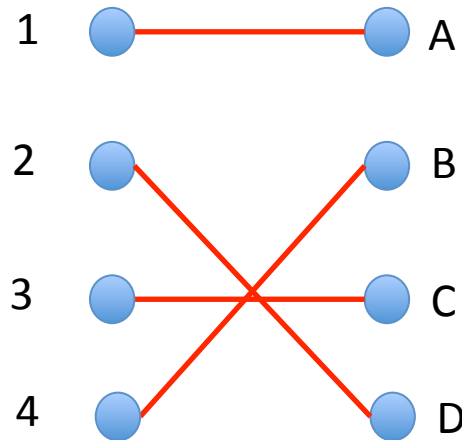


3 proposes to C.
C is free; so accepts.

Gale & Shapley's Greedy Algorithm (1962)

| Companies | Interns | | | |
|-----------|---------|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

| | | Companies | | | |
|---------|---|-----------|---|---|---|
| Interns | A | 1 | 3 | 2 | 4 |
| | B | 4 | 3 | 2 | 1 |
| | C | 2 | 3 | 1 | 4 |
| | D | 3 | 4 | 2 | 1 |



Is this matching stable?

Answer: Yes: B/c (1, A) mutually prefer each other most & every other company is matched to their 2nd best preference.

Gale & Shapley's Greedy Algorithm (1962)

```
procedure GS(n companies, n interns, and their rankings):  
  Initially all companies & interns are unmatched  
  while ( $\exists$  unmatched comp. who hasn't proposed to all interns):  
    c = pick one such company arbitrarily  
    i = highest ranked intern to whom c has not yet proposed  
    c "proposes" to i:  
    if (i is free): (c, i) are "tentative matched"  
    else {  
      c` = the company i is currently matched to  
      if (i prefer c over c`):  
        (c, i) are matched and c` is unmatched.  
    }  
  return all matched pairs
```

Q1: Will this algorithm always terminate?

Q2: If it terminates, will it return a matching?

Q3: If it returns a matching, is the matching stable?

Q4: If it terminates, does it always return the same matching?

Questions

1. Does GS terminate?
2. If it terminates, does GS return a matching?
3. If it terminates & returns a matching, is that matching stable?
4. If it terminates & returns a stable matching, does it always return the same stable matching?

Q1: Does GS Terminate? (1)

procedure GS(n companies, n interns, and their rankings):

Initially all companies & interns are unmatched

while (\exists unmatched comp. who hasn't proposed to all interns):

c = pick one such company arbitrarily

i = highest ranked intern to whom c has not yet proposed

c “proposes” to i :

if (i is free): (c, i) are “tentative matched”

else {

c' = the company i is currently matched to

if (i prefer c over c'):

 (c, i) are matched and c' is unmatched.

 }

return all matched pairs

Q: How many times can this while loop execute?

Simpler Q: How many times can GS pick the same company in the while loop?

Answer to Simpler Q: n (b/c each company proposes to n interns at most.)

Answer to Q: $O(n^2)$

Q1: Does GS Terminate? (4)

Claim: GS terminates after $O(n^2)$ while loop iterations.

Proof: At each while loop iteration, GS picks an unmatched company who hasn't yet proposed to every n intern.

GS can pick the same company at most n times.

There are n companies.

So there are at most n^2 while loop iterations.

Questions

1. Does GS terminate? Answer: Yes!
2. If it terminates, does GS return a matching?
3. If it terminates & returns a matching, is that matching stable?
4. If it terminates & returns a stable matching & does it always return the same stable matching?

Q2: Does GS Return a Matching? (1)

◆ Consider a free intern i that at some point is matched to a company.

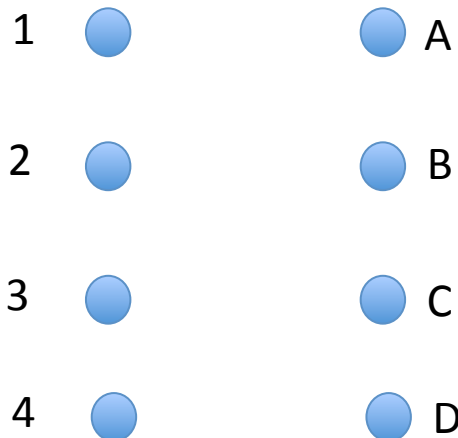
Q: Is i ever unmatched again?

A: No, i always remains matched (can only change her companies).

Q: Does i 's sequence of companies get better over time?

A: Yes, i only changes company from c to c' if i prefer c' better than c .

Observation 1: A matched intern i remains matched forever & i 's companies get better and better.



Q2: Does GS Return a Matching? (2)

◆ Consider a free company c that at some point is matched to an intern.

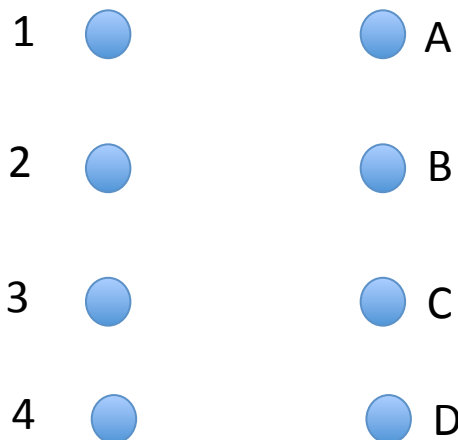
Q: Is c ever unmatched again?

A: Possibly, if c 's match leaves for another company.

Q: Does c 's sequence of interns get better over time?

A: No, it actually gets worse and worse over time.

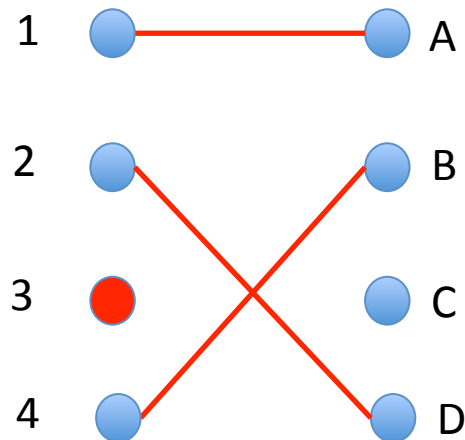
Observation 2: At each match, c proposes to an intern lower in c 's list.



Q2: Does GS Return a Matching? (3)

I.e: Can there be an unmatched company in the end?

Suppose 3 is unmatched in termination.



This means there is an intern who is not matched as well, say C.

⇒ At some point 3 did propose to C.

⇒ At that point C was either matched or unmatched.

⇒ Case 1: Matched => Contradiction: recall matched intern remains engaged.

⇒ Case 2: Unmatched => Contradiction, then was matched to 3 and must have remained matched.

Q2: Does GS Return a Matching? (For Your Notes)

I.e: Can there be an unmatched company in the end?

Hint: Recall Part of Observation 1: Whenever a free intern i gets matched, i remains matched forever.

Claim: No

Proof: By contradiction. Suppose a company c remained unmatched.

This means at termination, c already proposed to every intern.

This means that every intern i rejected c either:

1. Because i was already matched to another company when c proposed;
2. i got matched to c , but then left c .

In either case, by Observation 1 i was matched at termination.

This is a contradiction because if every intern is matched, then one of them must be matched to c .

Q.E.D.

Questions

1. Does GS terminate? Answer: Yes!
2. If it terminates, does GS return a matching? Answer: Yes!
3. If it terminates & returns a matching, is that matching stable?
4. If it terminates & returns a stable matching & does it always return the same stable matching?

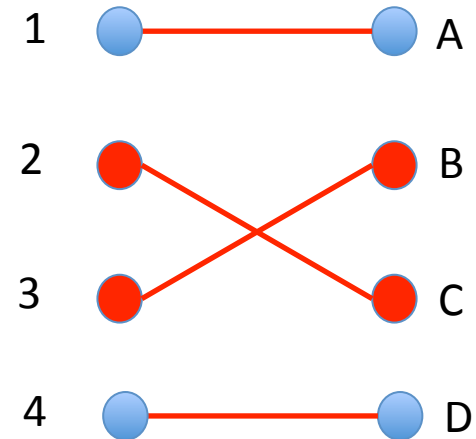
Q3: Does GS Return a *Stable* Matching? (1)

I.e: Can there be an unstable pair (c, i) (c', i') ?

Suppose $(2, C)$ and $(3, B)$ are unstable.

Suppose w.l.o.g 2 prefers $B > C$ and B prefers $2 > 3$.

Why can't this happen?



B/c 2 proposed to B before C but was either:

(1) Rejected immediately \Rightarrow B was with a company B prefers more than 2.

\Rightarrow But this cannot happen b/c by Observation 1, B's companies get better and better, so B couldn't end with 3 (which is ranked below 2).

(2) Got matched but later left for another company B ranks higher.

\Rightarrow Cannot happen by the same argument.

Q3: Does GS Return a *Stable* Matching? (For Your Notes)

I.e: Can there be an unstable pair (c, i) (c', i') ?

Claim: No

Proof: By contradiction

Suppose w.l.o.g. c prefers $i' > i$ and i' prefers $c > c'$.

This means at some point in time t , c proposed to i' but i' either:

(1) rejected c b/c i' was matched to c'' who was higher ranked (according to i' 's preferences).

=> Then i' ranks $c'' > c > c'$

=> Contradiction b/c by Observation 1 i' 's companies only get better.

(2) got matched to c but left c for another company c'' .

=> Contradiction by the same argument above.

Q.E.D.

Questions

1. Does GS terminate? Answer: Yes!
2. If it terminates, does GS return a matching? Answer: Yes!
3. If it terminates & returns a matching, is that matching stable

Answer: Yes!

4. If it terminates & returns a stable matching & does it always return the same stable matching?

What Have We Proved So Far?

We effectively proved that no matter what the preferences are:

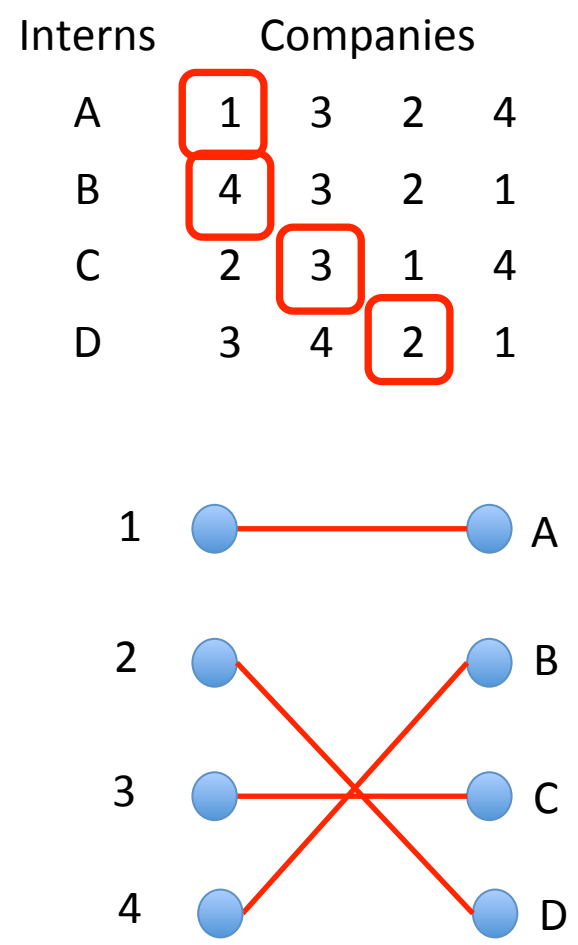
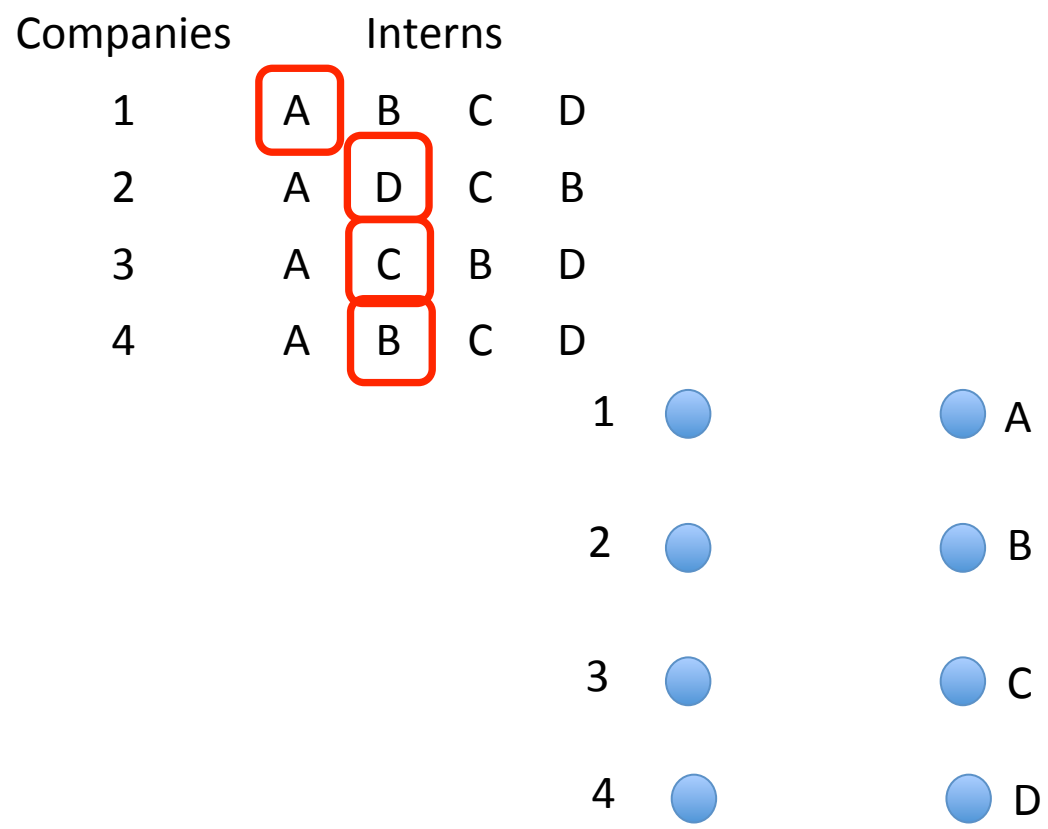
1. There is always a stable matching.
2. We can find it efficiently. ($O(n^2)$ time).

Why?

****B/c GS returns one!!****

Q4: Does GS Return The Same *Stable* Matching?

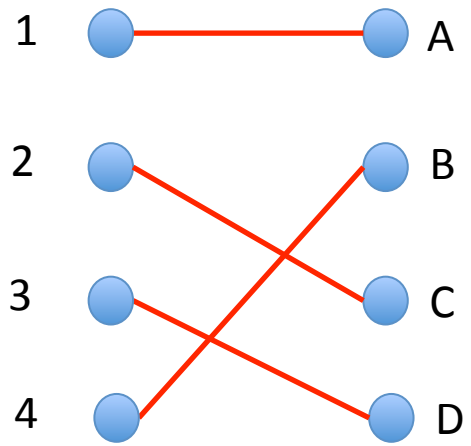
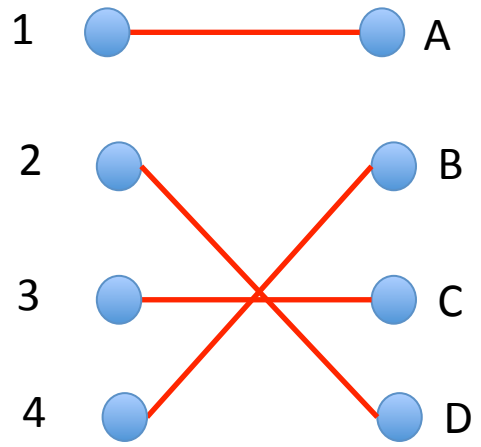
- ◆ Recall we can pick any unmatched company in the while loop.
- ◆ Does this matter?



Q4: Does GS Return The Same *Stable* Matching?

| Companies | Interns | | | |
|-----------|---------|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

| Interns | Companies | | | |
|---------|-----------|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



Stable Matching 1
** Our 2 simulations of GS
returned this one **

Stable Matching 2

Looks like GS favors companies?!

What if Interns Proposed?

Companies

Interns

1

A

B

C

D

2

A

D

C

B

3

A

C

B

D

4

A

B

C

D

1

2

3

4

1

2

3

4

A

B

C

D

Interns

Companies

A

1

3

2

4

B

4

3

2

1

C

2

3

1

4

D

3

4

2

1

What if Interns Proposed?

Companies

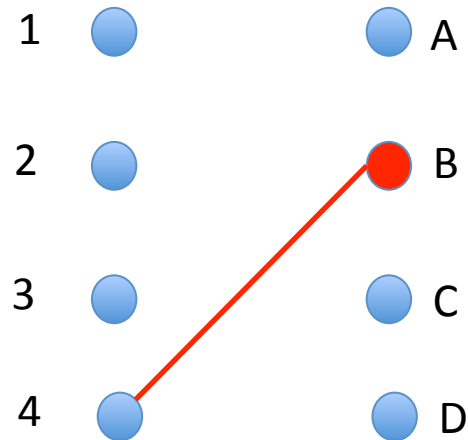
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



What if Interns Proposed?

Companies

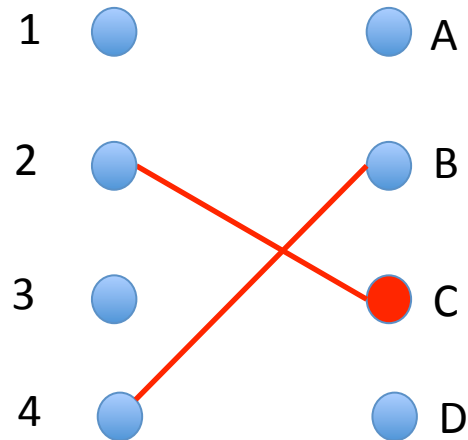
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



What if Interns Proposed?

Companies

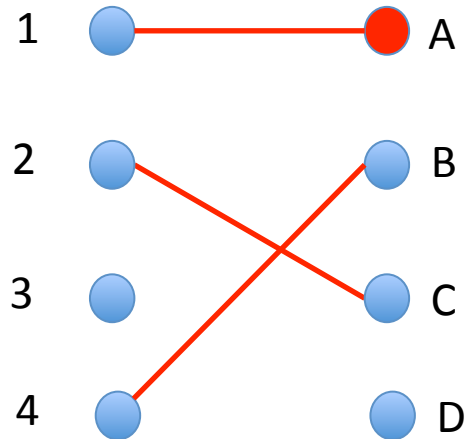
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



What if Interns Proposed?

Companies

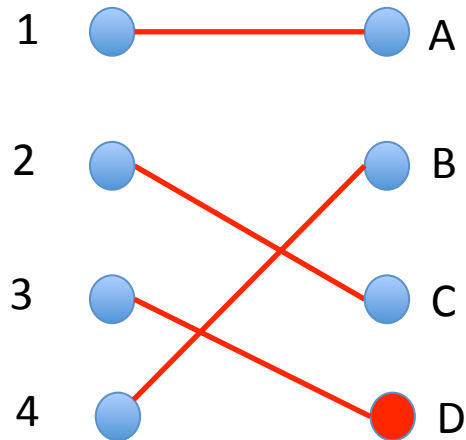
Interns

| | | | | |
|---|---|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |

Interns

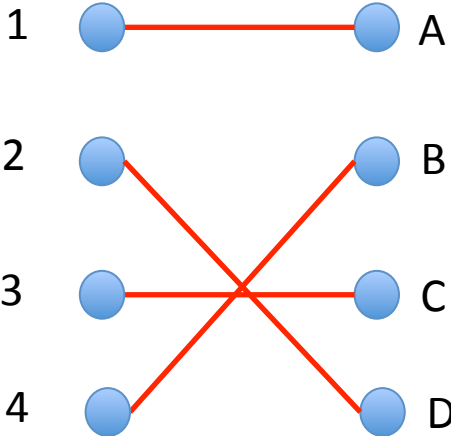
Companies

| | | | | |
|---|---|---|---|---|
| A | 1 | 3 | 2 | 4 |
| B | 4 | 3 | 2 | 1 |
| C | 2 | 3 | 1 | 4 |
| D | 3 | 4 | 2 | 1 |



What if Interns Proposed?

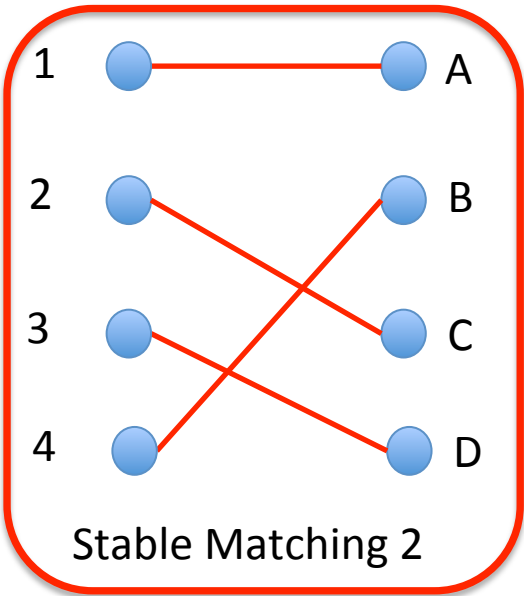
| Companies | Interns | | | |
|-----------|---------|---|---|---|
| 1 | A | B | C | D |
| 2 | A | D | C | B |
| 3 | A | C | B | D |
| 4 | A | B | C | D |



Stable Matching 1

** Our 2 simulations of GS returned this one **

| | | Interns | Companies | | | |
|---|---|---------|-----------|---|--|--|
| A | 1 | 3 | 2 | 4 | | |
| B | 4 | 3 | 2 | 1 | | |
| C | 2 | 3 | 1 | 4 | | |
| D | 3 | 4 | 2 | 1 | | |



Stable Matching 2

Now we got the 2nd stable matching (which is better for interns).

Does GS Favor Companies?

Dfn (valid company/intern): An intern i is valid for c iff \exists a stable matching in which c is matched with i . (and vice versa for interns)

Dfn (optimal company/intern): An intern i is optimal for c iff c ranks i highest among all valid interns for c .

Dfn (pessimal company/intern): An intern i is pessimal for c iff c ranks i lowest among all valid interns for c .

Claim 1: Every company c is matched c 's optimal intern by GS!

Claim 2: Every intern i is matched to i 's pessimal company by GS!

Claim: Every c matches c 's optimal intern by GS.

First note that this claims that the following is a matching.

Let $\text{optimal}(c)$ be the optimal intern for company c .

1, $\text{optimal}(1)$

2, $\text{optimal}(2)$

3, $\text{optimal}(3)$

...

n , $\text{optimal}(n)$

Moreover, if the claim is true: $\text{optimal}(c) \neq \text{optimal}(c')$ for any c and c' .

Proof Sketch of Claim

Suppose during an execution E of GS a company c is rejected by $\text{optimal}(c)$.

Consider the first time this happens and w.l.o.g it happens to company 1 when proposing to D : I.e. suppose $\text{optimal}(1)$ is D .

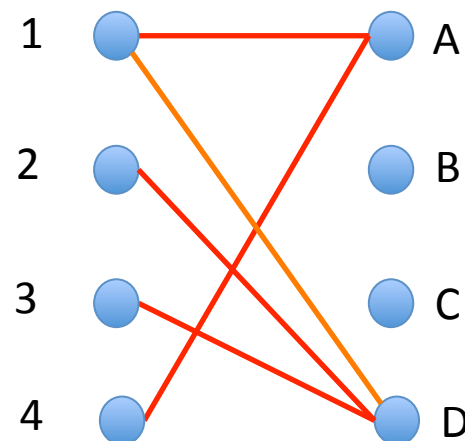
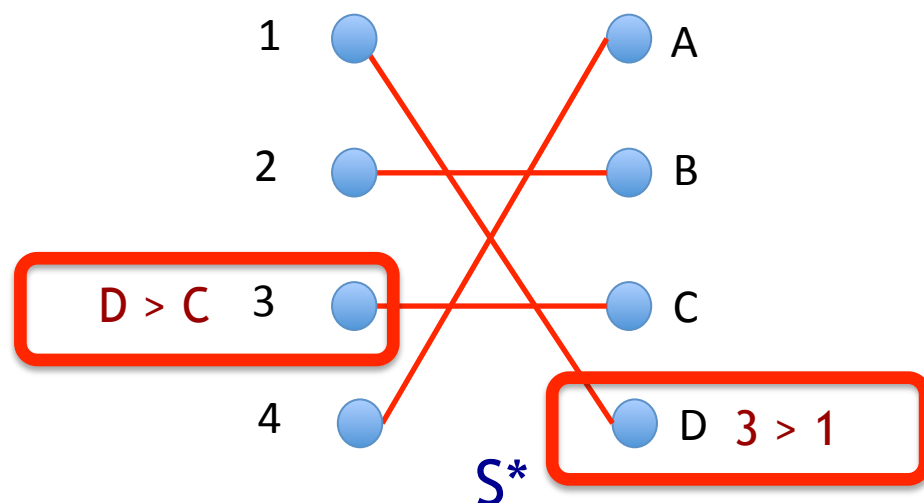
But D rejected 1 (say for 3).

By dfn of “optimal” intern \exists a stable matching S^* with $(1, D)$ is matched.

Then D prefers $3 > 1$.

Also 3 prefers $D > C$ (b/c 1 was the first company c rejected by $\text{optimal}(c)$)

$\Rightarrow (1, D) (3, C)$ is not stable! Contradiction.



Q.E.D.

Q4: Does GS Return The Same *Stable* Matching?

- ◆ Yes! It returns the company-optimal one.
- ◆ Exercise: Prove Claim 2 that what GS returns is also inter-pessimal.
- ◆ Moral: Make your moves first!

Questions

1. Does GS terminate? Answer: Yes!
2. If it terminates, does GS return a matching? Answer: Yes!
3. If it terminates & returns a matching, is that matching stable

Answer: Yes!

4. If it terminates & returns a stable matching & does it always return the same stable matching?

Answer: Yes!

The Residency Match

- ◆ Matching Medical students and residency slots (internships) in hospitals.
- ◆ > 100 years old program
- ◆ Internships are cheap: Medical schools want students badly!
- ◆ Before 1940s: Offers are made in senior years.
- ◆ Early 1940s: Junior years
- ◆ Mid 1940s: Some hospitals made offers as early as sophomore years.
- ◆ 1950s: National Residency Matching Program (NRMP) formed
 - ◆ Centralized system: Students rank hospitals; Hospitals rank students
 - ◆ NPMR produces a matching (but was not stable!)
 - ◆ 1952: Traditional propose & reject model => stable (shown in 1962)
- ◆ Until 2000s hospitals proposed => favoring hospitals
- ◆ About 10 years ago. Students started proposing.