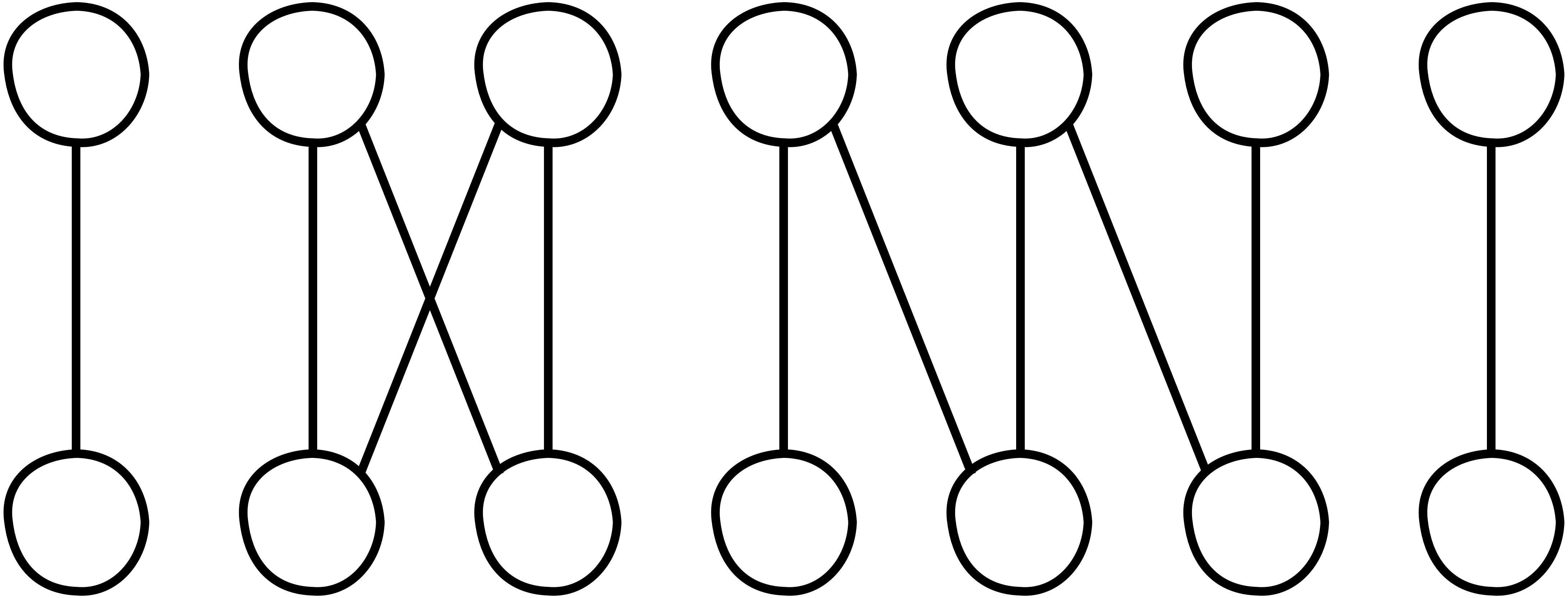
Q: How many perfect matchings in this graph?



CSE 331:
Algorithms & Complexity
"Gale-Shapley III"

Prof. Charlie Anne Carlson (She/Her)

Lecture 6

Wednesday September 9th, 2025

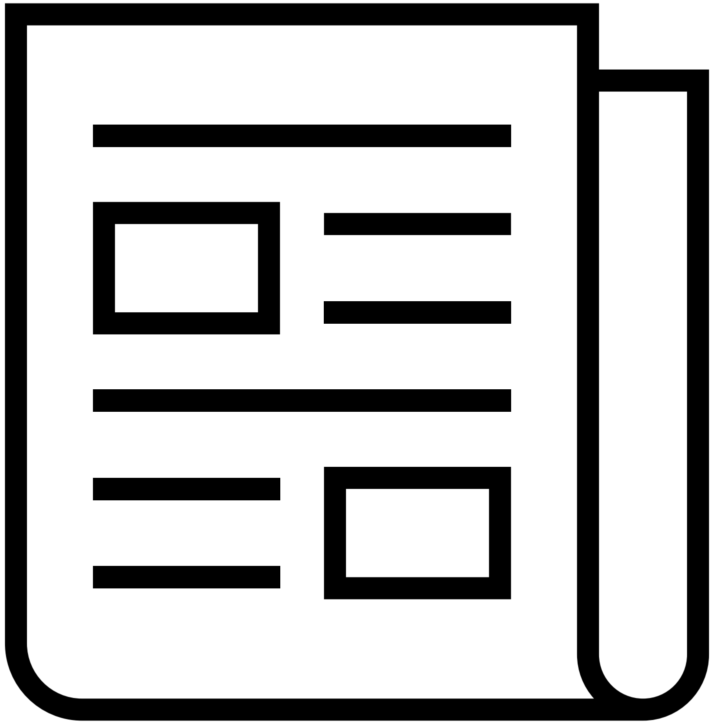
University at Buffalo

Schedule

1. Course Updates
2. Recap
3. Even More Gale-Shapely

Course Updates

- Complete Syllabus Quiz
- HW 1 Posted
  - Autolab Tomorrow
- HW 0 Being Graded
- Website Moved
- Project Signup before 19th



Gale-Shapley Algorithm

GALE–SHAPLEY (*preference lists for hospitals and students*)

INITIALIZE M to empty matching.

WHILE (some hospital h is unmatched and hasn't proposed to every student)

  $s \leftarrow$ first student on h's list to whom h has not yet proposed.

  IF (s is unmatched)

    Add h–s to matching M.

  ELSE IF (s prefers h to current partner h')

    Replace h'–s with h–s in matching M.

  ELSE

    s rejects h.

RETURN stable matching M.

Claim Time!

Claim: The algorithm always outputs a stable matching.

WHILE (some hospital h is unmatched and hasn't proposed to every student)

$s \leftarrow$ first student on h's list to whom h has not yet proposed.

IF (s is unmatched)

Add h–s to matching M.

ELSE IF (s prefers h to current partner h')

Replace h'–s with h–s in matching M.

ELSE

s rejects h.

Proof Idea Time!

Claim: The algorithm always outputs a stable matching.

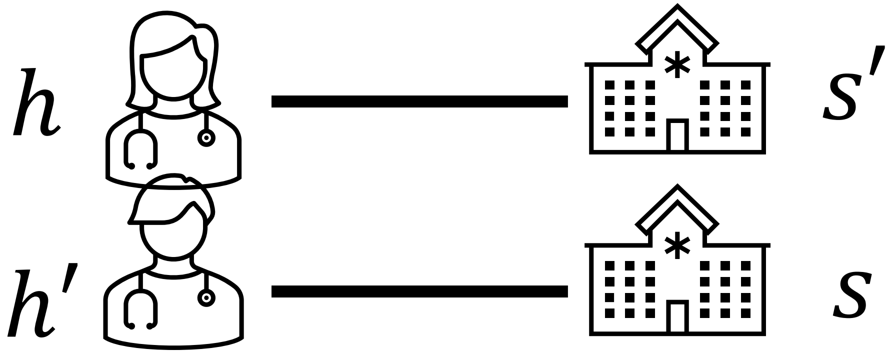
Proof (Ideas):

1. Show that every unmatched pair is not unstable.
2. **Show that the existence of an unstable pair would be a contradiction.**

Proof Time!

Proof: We proceed with a proof by contradiction.

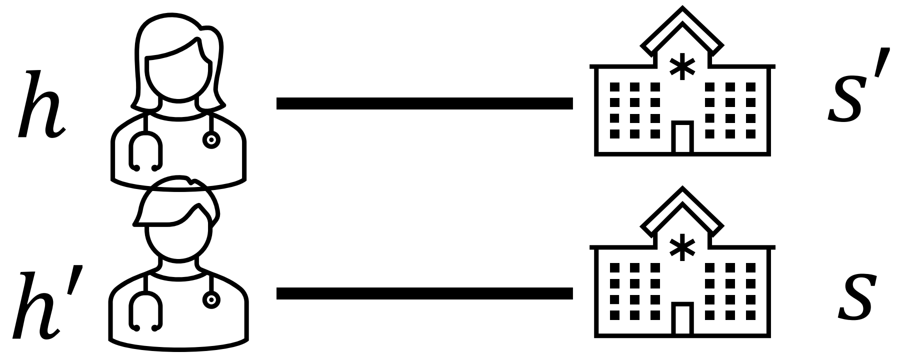
- Suppose to the contrary that the matching M returned was unstable. Then there must exist an unstable pair (h, s).
  - Since we know that matching returned is perfect, we know there exists hospital h' and student s' such that $(h, s') \in M$ and $(h', s) \in M$.



Proof Time!

Proof: …

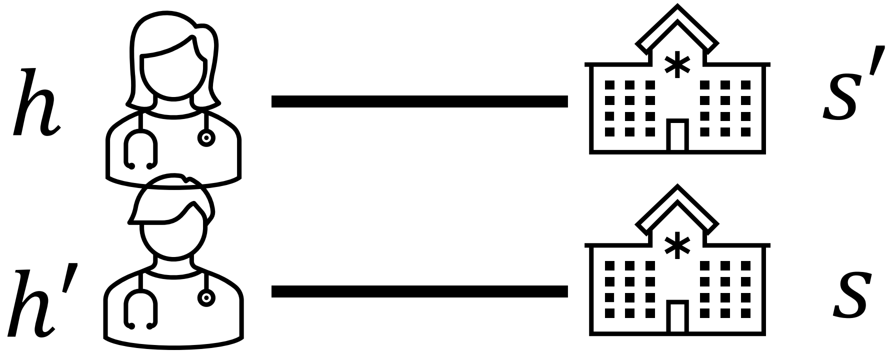
- Suppose h did propose to s at some point in the algorithm.
  - Then s must have at some point rejected h and thus, must prefer h' to h. (Not Unstable)



Proof Time!

Proof: …

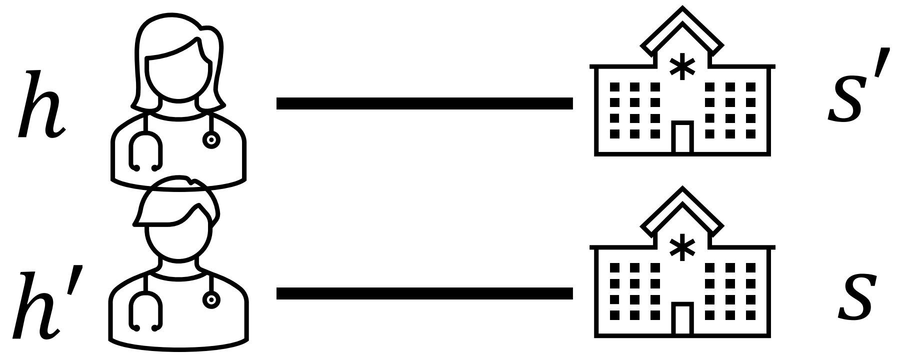
- Suppose h did not propose to s at some point in the algorithm.
  - Then h must have proposed to s' first and thus, must prefer s' to s. (Not Unstable)



Proof Time!

Proof: …

- In all cases we have concluded that if there is an unstable pair then the pair is can't be unstable. This is a contradiction and thus there can be no unstable pairs.

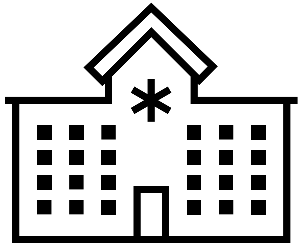


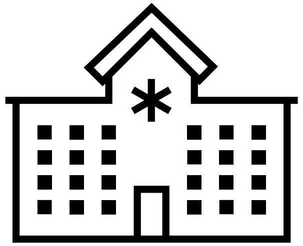
Q: Is there just one Stable Matching?

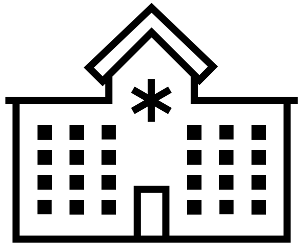
Dr. One
A > B > C


Dr. Two
B > A > C

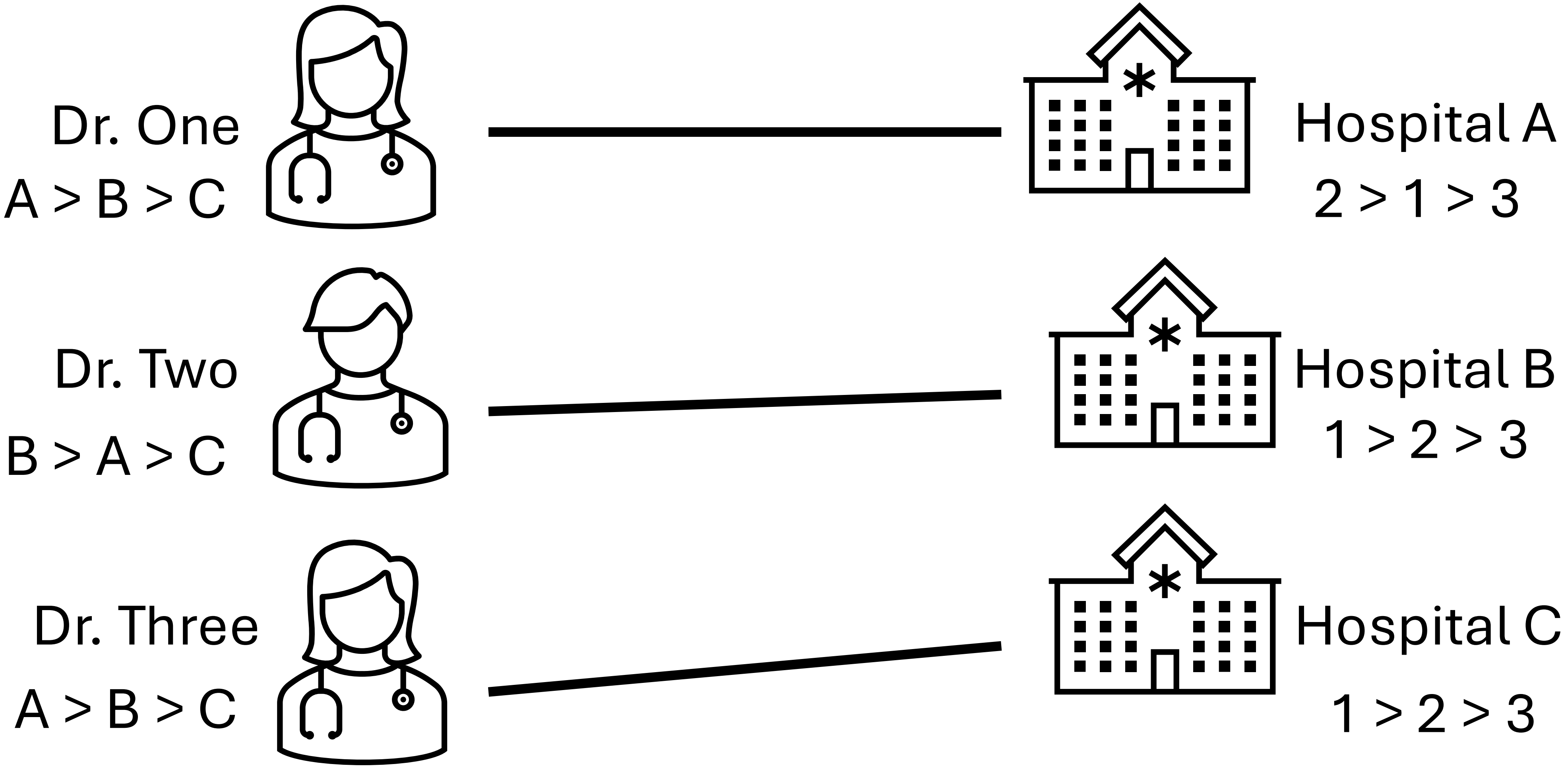

Dr. Three
A > B > C



Hospital A
2 > 1 > 3

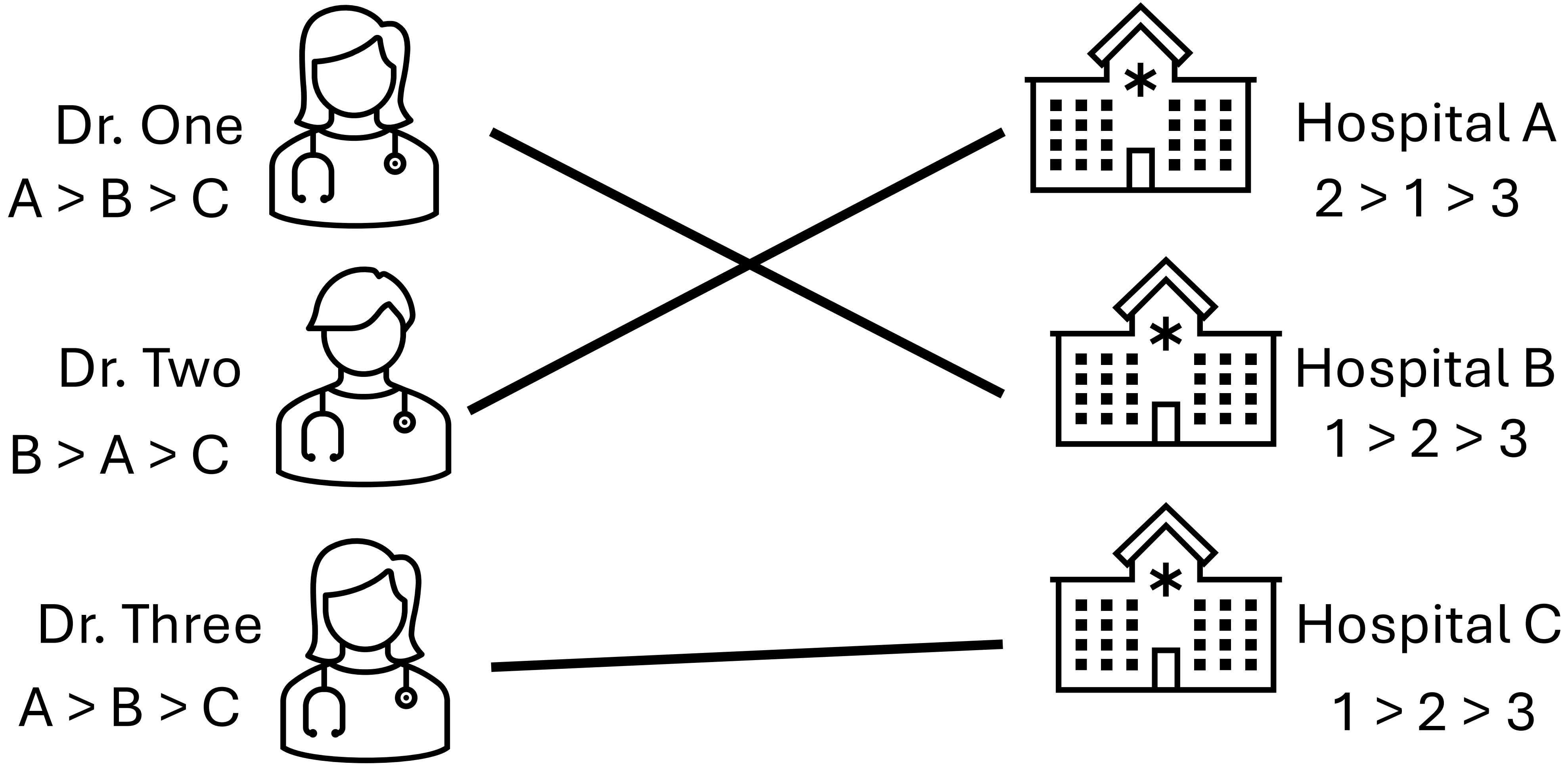

Hospital B
1 > 2 > 3


Hospital C
1 > 2 > 3

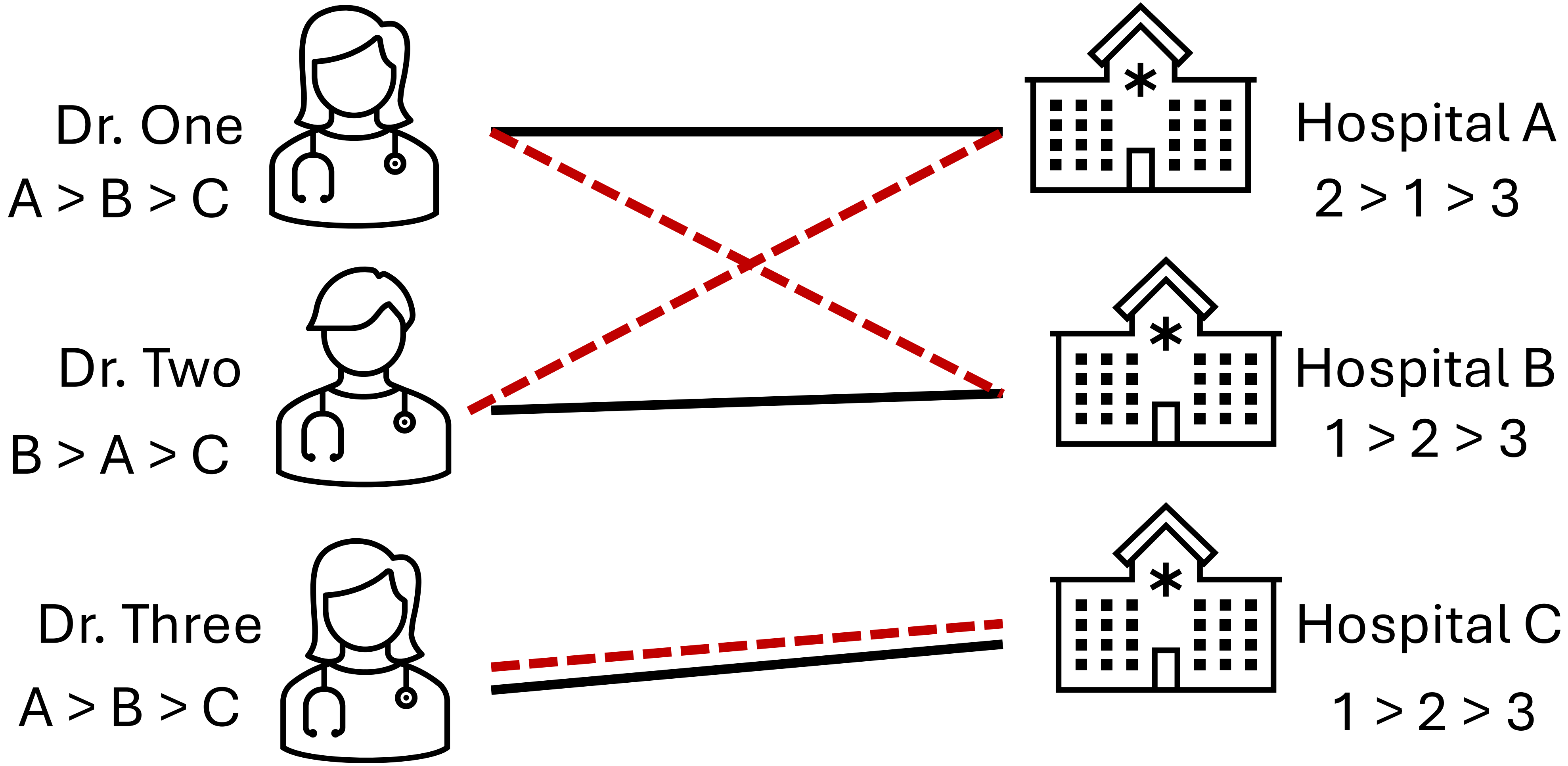
Matching I


Dr. One
A > B > C
Dr. Two
B > A > C
Dr. Three
A > B > C
Hospital A
2 > 1 > 3
Hospital B
1 > 2 > 3
Hospital C
1 > 2 > 3

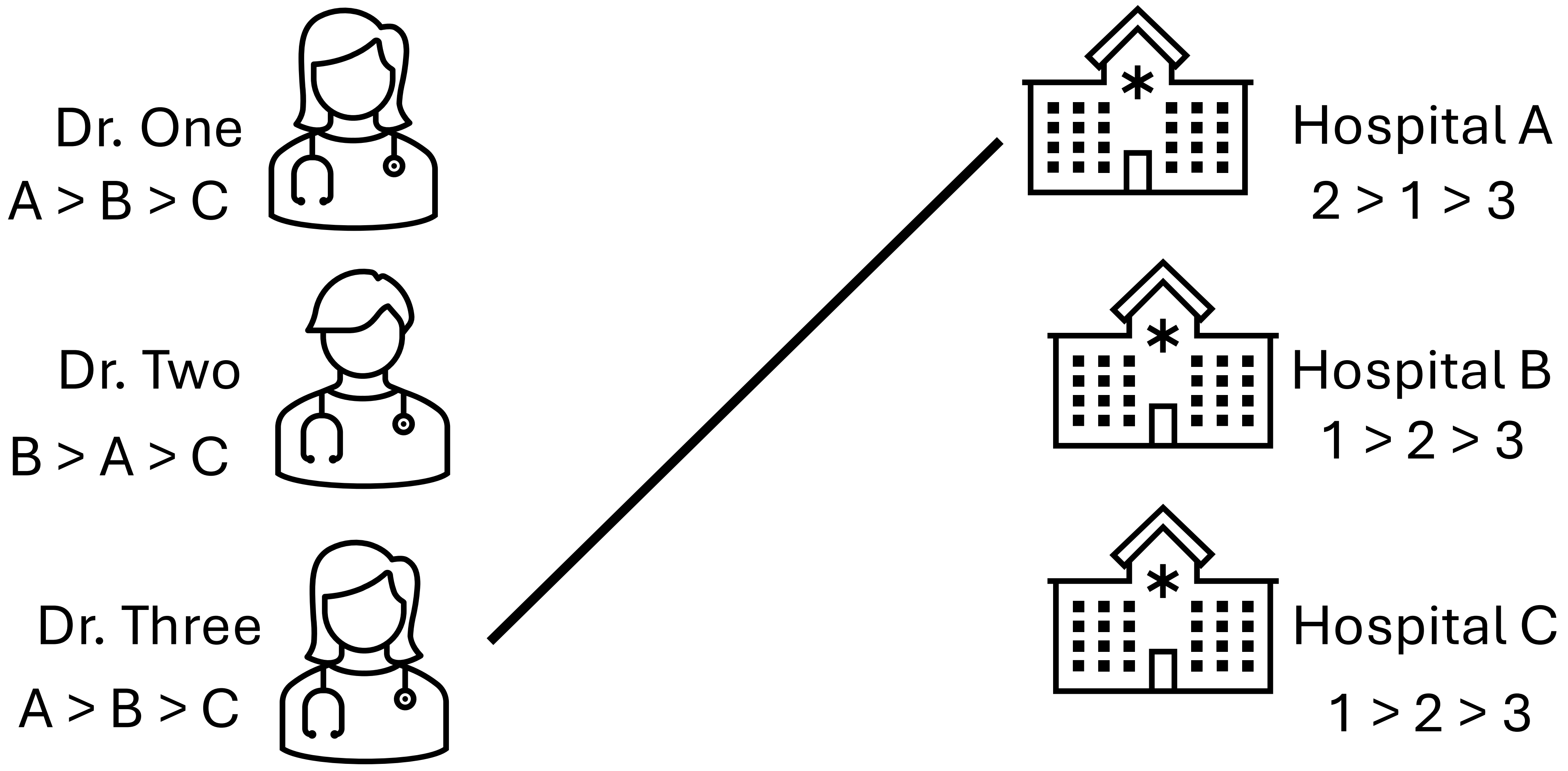
Matching II


Dr. One
A > B > C
Dr. Two
B > A > C
Dr. Three
A > B > C
Hospital A
2 > 1 > 3
Hospital B
1 > 2 > 3
Hospital C
1 > 2 > 3

Matching I vs Matching II



Q: Is there a Stable Matching with this edge?

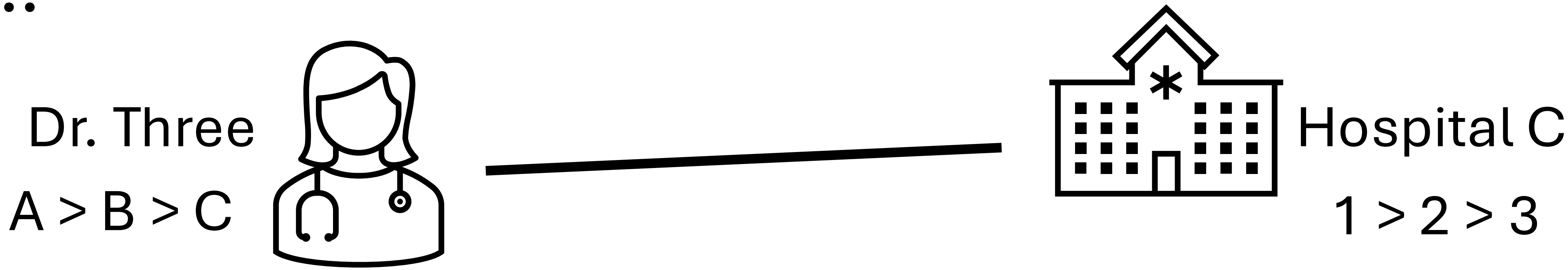


Valid Partners

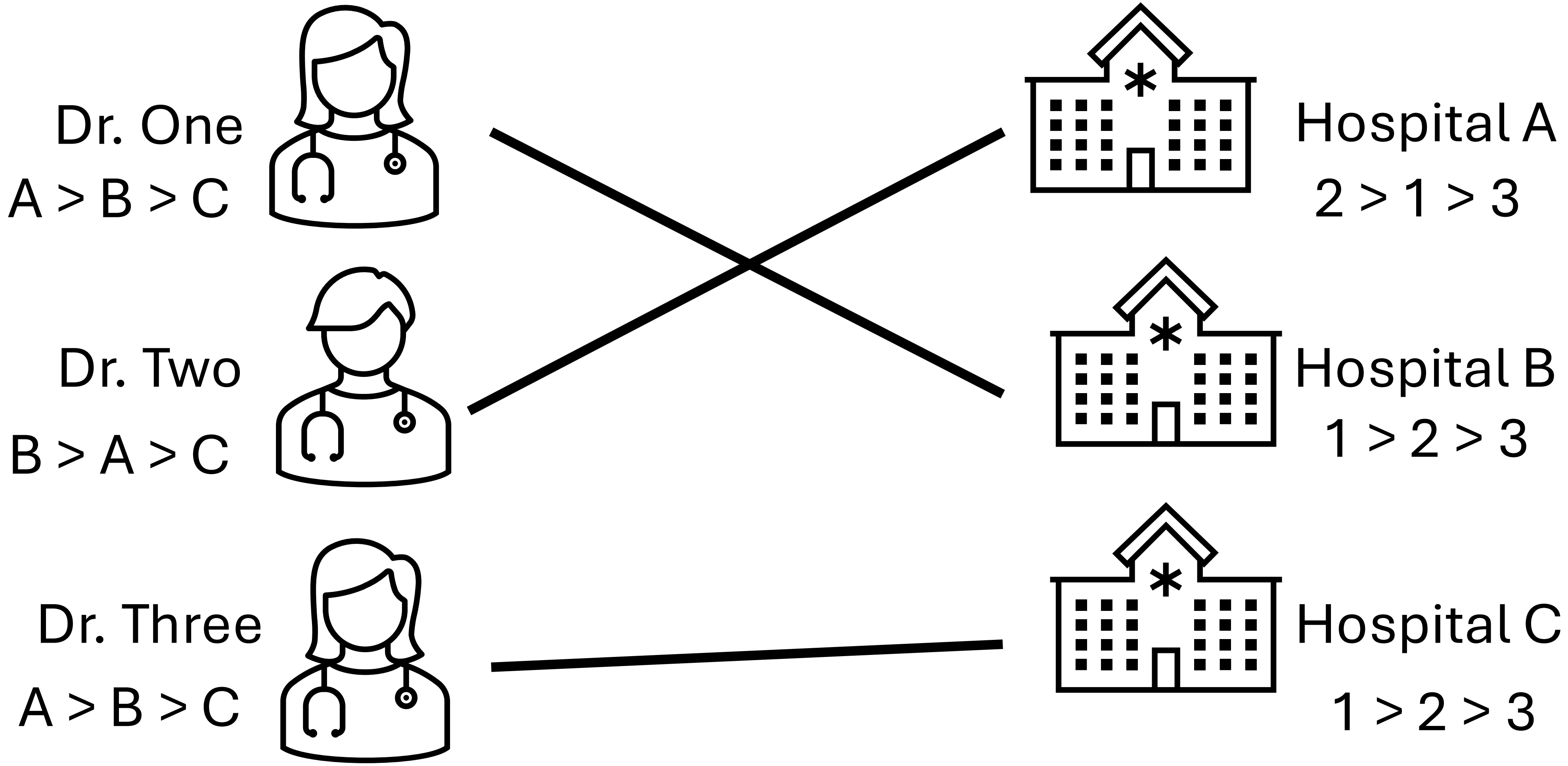
Def: We say student s is a *valid partner* of hospital h if there exists a stable matching in which s and h are matched.

Q: What would be a student's *best valid partner*?

E.g.:



Q: Who is Dr. Two's best valid partner?



Valid Partners

Def: The *student optimal assignment* is the one in which all students are matched with their best valid partner. The *hospital optimal assignment* is the one in which all hospitals are matched with their best valid partner.

Q: Are these optimal assignments stable matchings?
Q: Can we find them?

Claim Time!

Claim: The algorithm always outputs the hospital-optimal assignment.

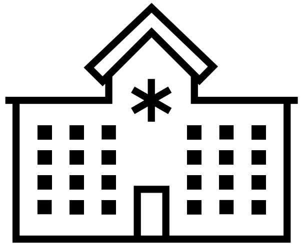
```
WHILE (some hospital h is unmatched and hasn't proposed to every student)
  s ← first student on h's list to whom h has not yet proposed.
  IF (s is unmatched)
    Add h–s to matching M.
  ELSE IF (s prefers h to current partner h′)
    Replace h′–s with h–s in matching M.
  ELSE
    s rejects h.
```

Proof Idea Time!

Claim: The algorithm always outputs a stable matching.

Proof:

- Suppose to the contrary that the output is not hospital optimal. We will show that….



Proof Idea Time!

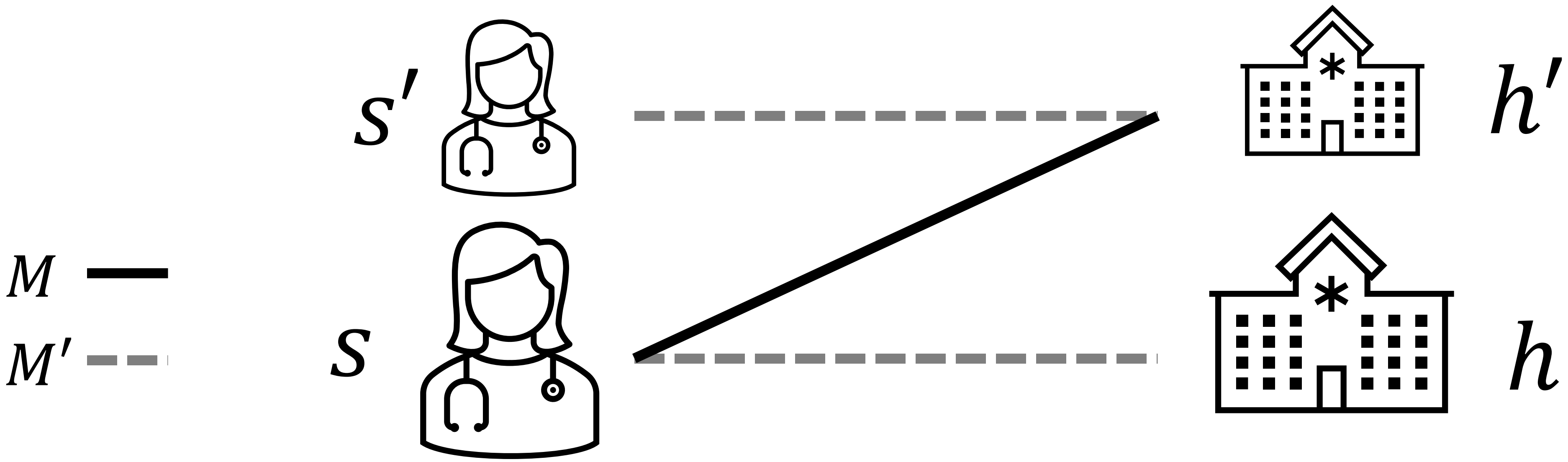
Proof:

- Suppose to the contrary that the output M is not hospital optimal.
- Then let h be the **first hospital** to be rejected by a valid partner s.
- Let M' be the stable matching with s and h paired together.

Proof Idea Time!

Proof: …

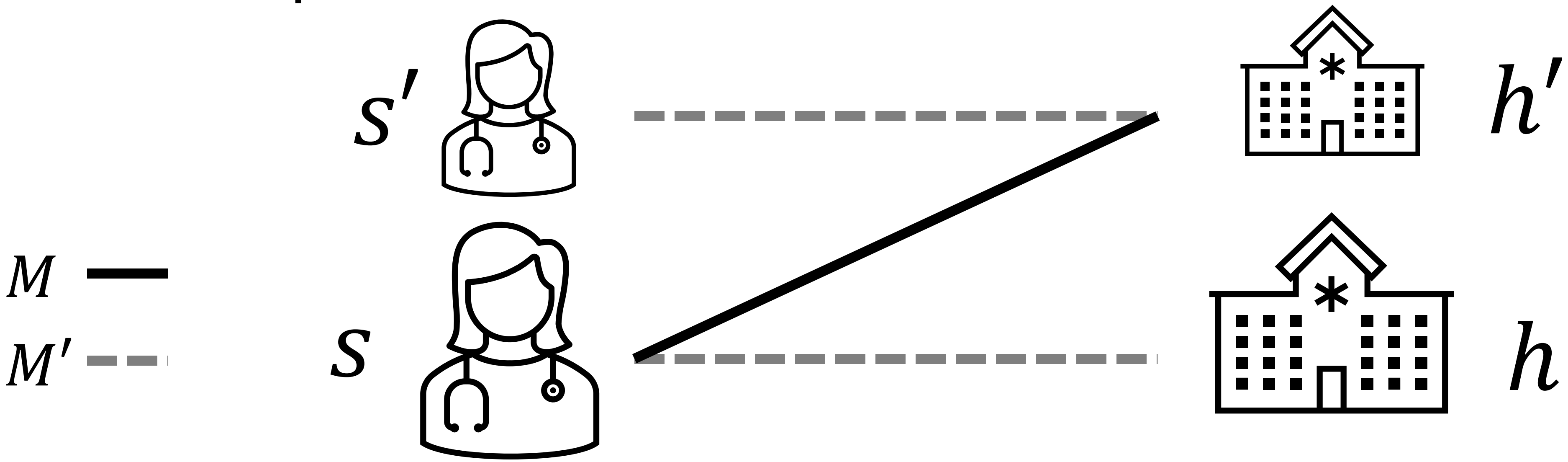
- Let h' be the hospital paired with s in the output of the algorithm. Let s' be the student paired with h' in M.



Proof Idea Time!

Proof: …

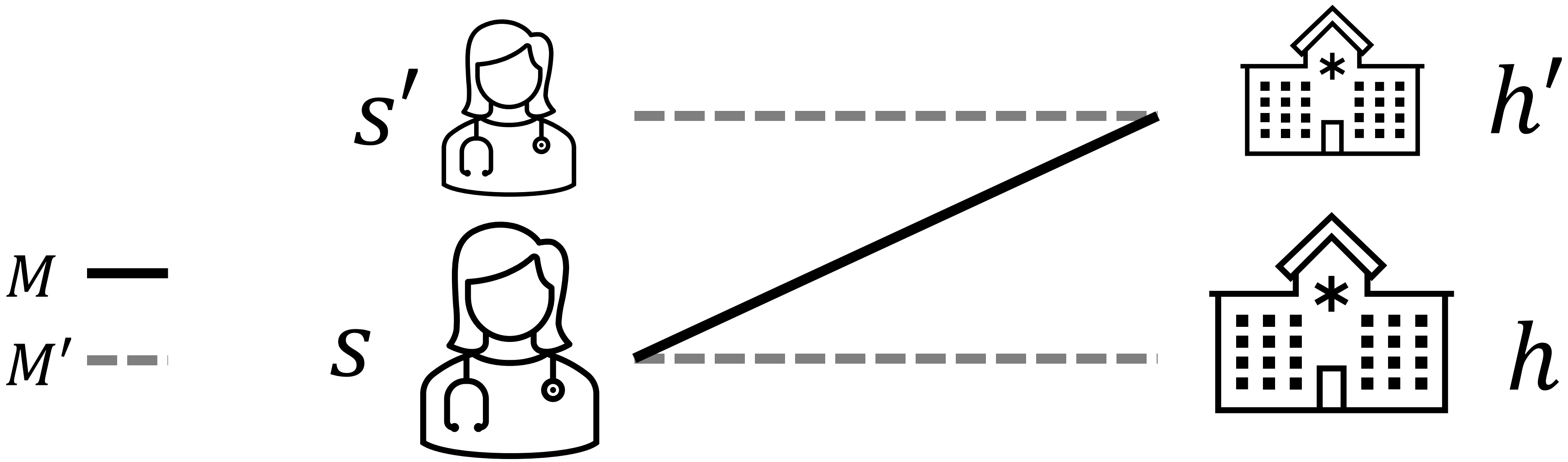
- Since h was rejected by s, it means that s prefers h' over h
- Since h was the **first** rejected, h' did not get rejected and must prefer s to s'



Proof Idea Time!

Proof: …

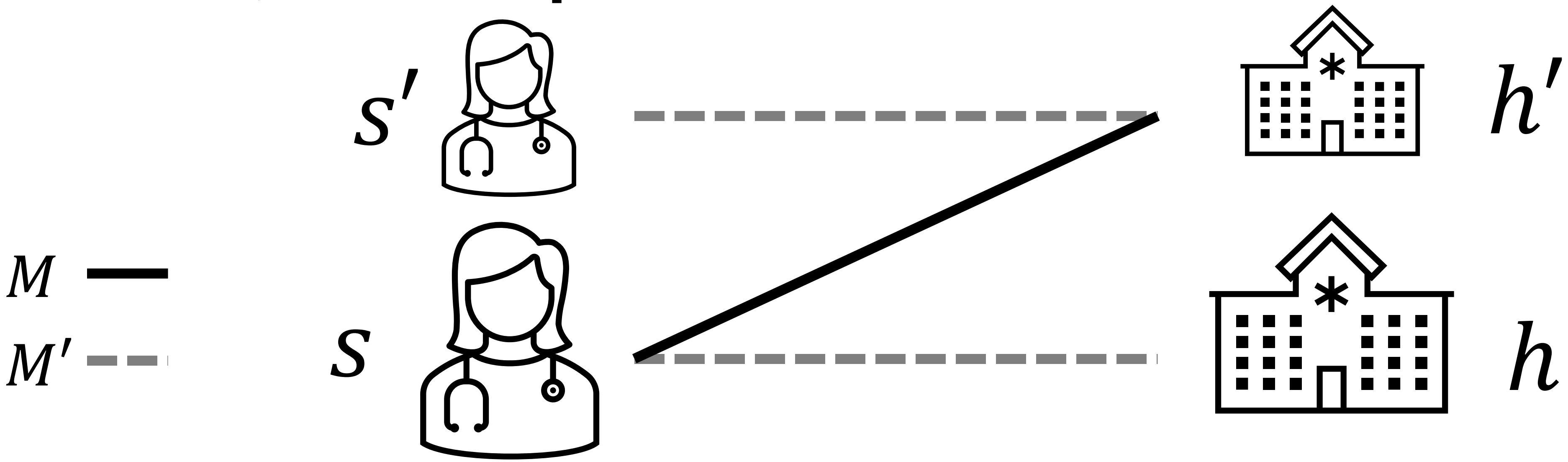
- Since h was rejected by s, it means that **s prefers h' over h**



Proof Idea Time!

Proof: …

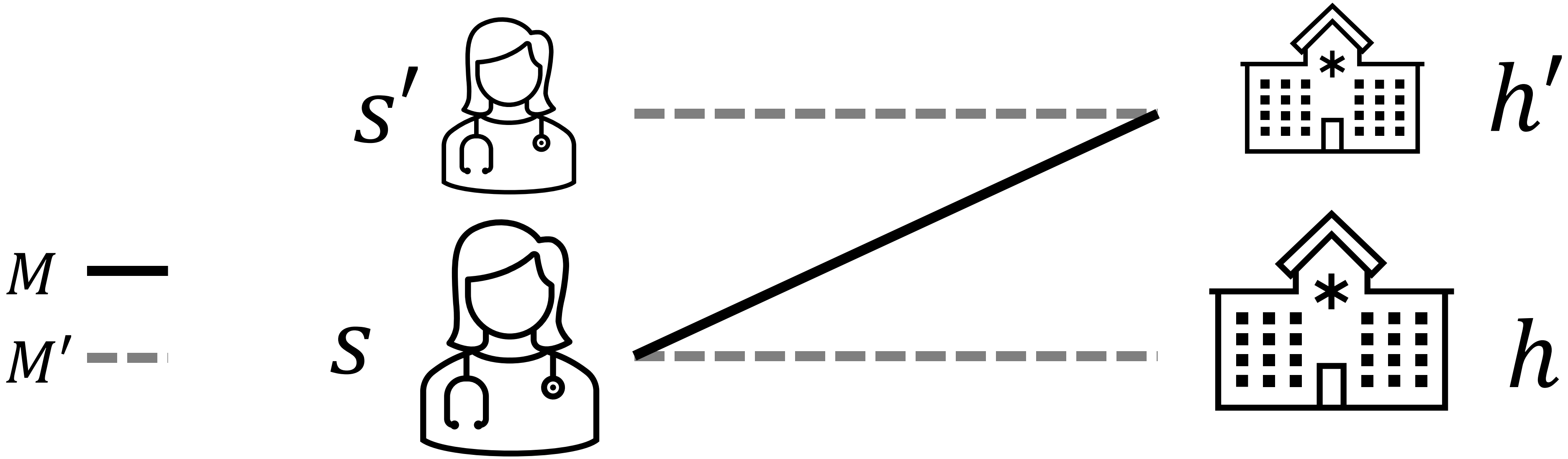
- Since h was rejected by s, it means that $\boldsymbol{s}$ **prefers** $\boldsymbol{h'}$ **over** $\boldsymbol{h}$
- Since h was the first rejected, h' did not get rejected and thus, $\boldsymbol{h'}$ **must prefer** $\boldsymbol{s}$ **to** $\boldsymbol{s'}$



Proof Idea Time!

Proof: …

- If s **prefers** $\boldsymbol{h'}$ **over** $\boldsymbol{h}$ and $\boldsymbol{h'}$ **must prefer** $\boldsymbol{s}$ **to** $\boldsymbol{s'}$ then (h', s) is an unstable pair in M'. This is a contradiction!



Claim Time!

Claim: The algorithm always outputs the *student-pessimal* assignment.

WHILE (some hospital h is unmatched and hasn't proposed to every student)

$s \leftarrow$ first student on h's list to whom h has not yet proposed.

IF (s is unmatched)

Add h–s to matching M.

ELSE IF (s prefers h to current partner h')

Replace h'–s with h–s in matching M.

ELSE

s rejects h.

Claim Time!

Each student gets their worst valid partner.

Claim: The algorithm always outputs the *student-pessimal* assignment.

WHILE (some hospital h is unmatched and hasn't proposed to every student)

$s \leftarrow$ first student on h's list to whom h has not yet proposed.

IF (s is unmatched)

Add h–s to matching M.

ELSE IF (s prefers h to current partner h')

Replace h'–s with h–s in matching M.

ELSE

s rejects h.

Proof Idea Time!

Claim: The algorithm always outputs a stable matching.

Proof:

- Suppose to the contrary that the output M is not **student-pessimal**.
  - Then there exists student s that is not matched with their worst valid partner h in M.

Proof Idea Time!

Claim: The algorithm always outputs a stable matching.

Proof:

- Suppose to the contrary that the output M is not **student-pessimal**.
  - Then there exists student s that is not matched with their worst valid partner h in M.
  - Let M' be a stable matching such that s and h are matched.

Proof Idea Time!

Proof:

- Suppose to the contrary that the output M is not **student-pessimal**.
  - Then there exists student s that is not matched with their worst valid partner h in M.
  - Let M' be a stable matching such that s and h are matched.
- Let h' be the hospital s is matched to in M.

Proof Idea Time!

Proof:

- Suppose to the contrary that the output M is not **student-pessimal**.
  - Then there exists student s that is not matched with their worst valid partner h in M.
  - Let M' be a stable matching such that s and h are matched.
- Let h' be the hospital s is matched to in M.
  - Then s prefers h' to h.

Proof Idea Time!

Proof:

- Let h' be the hospital s is matched to in M.
  - Then s prefers h' to h otherwise h would not be worst valid pair as assumed.
- Let s' be the hospital h is matched to in M.

Proof Idea Time!

Proof:

- Let h' be the hospital s is matched to in M.
  - Then s prefers h' to h otherwise h would not be worst valid pair as assumed.
- Let s' be the hospital h is matched to in M.
  - Since M is hospital-optimal, h prefers s to s'.

Proof Idea Time!

Proof:

- Let h' be the hospital s is matched to in M.
  - Then s prefers h' to h otherwise h would not be worst valid pair as assumed.
- Let s' be the hospital h is matched to in M.
  - Since M is hospital-optimal, h prefers s to s'.
- Thus, (h, s) is an unstable pair in M. **<- Contradiction!**

Q: Does it Help to Lie?

Dr. One
C > A > B

Dr. Two
A > C > B

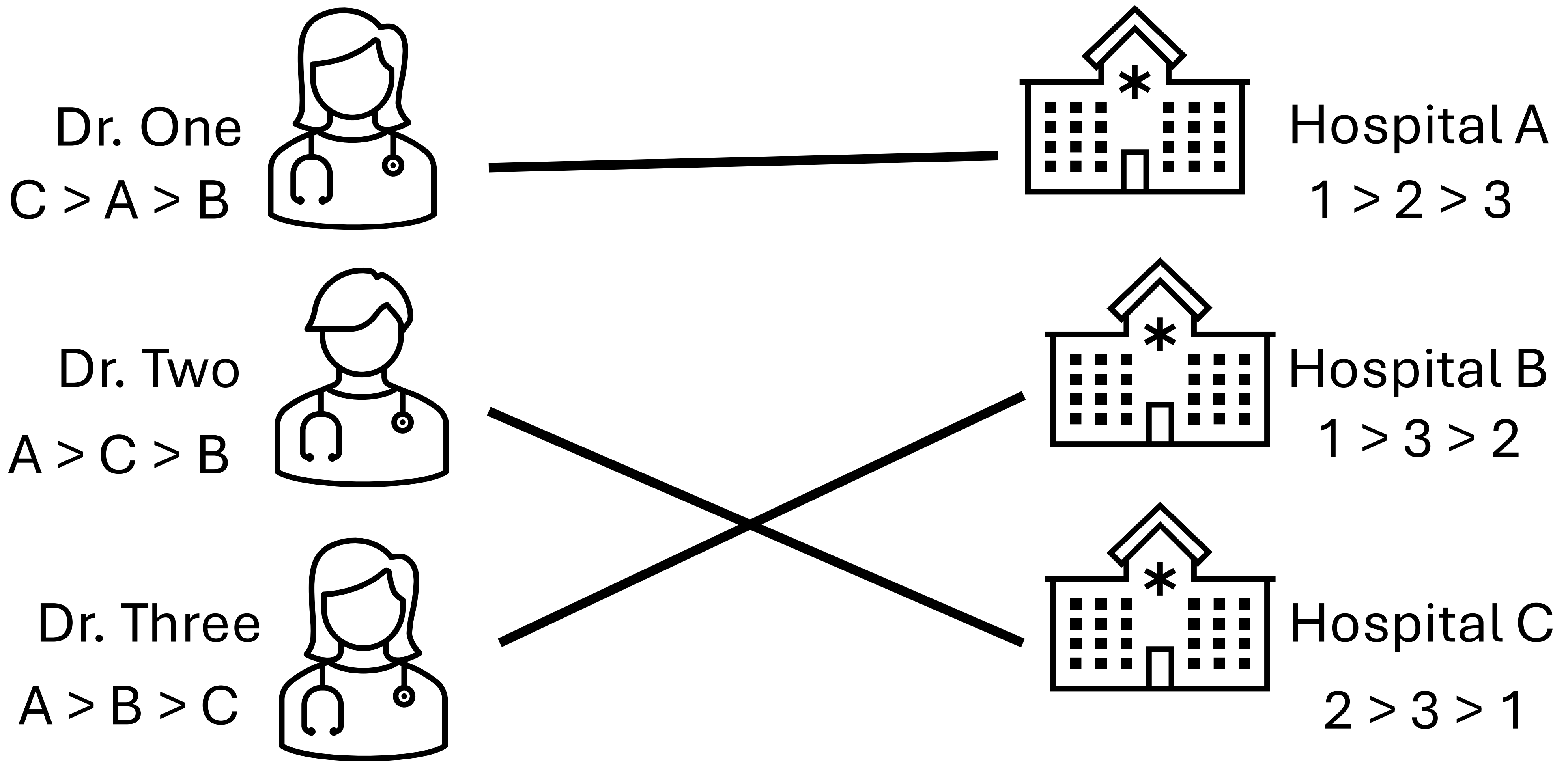
Dr. Three
A > B > C

Hospital A
1 > 2 > 3

Hospital B
1 > 3 > 2

Hospital C
2 > 3 > 1

Q: Does it Help to Lie?

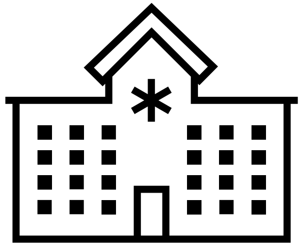


Q: Does it Help to Lie?

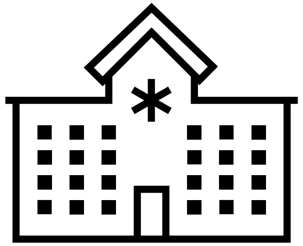
Dr. One
C > “**B > A**”

Dr. Two
A > C > B

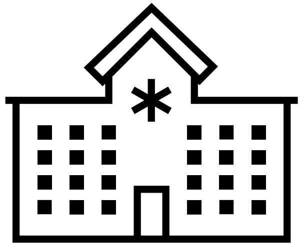
Dr. Three
A > B > C



Hospital A
1 > 2 > 3



Hospital B
1 > 3 > 2



Hospital C
2 > 3 > 1

Q: Does it Help to Lie?