# The Stable Marriage problem

Marriages made by algorithms,
guaranteed with stability !

Souvik Parial
IIT Guwahati

# Outline of the talk

- Stable Marriage problem
- Gale-Shapley algorithm
- Structure of Stable Matchings
- Application: Hospitals-Residents problem

# Stable Marriage(SM) problem

- Two disjoint sets **M**(men) and **W**(women), each of size **n**
- Each person has a strictly ordered preference list of **all** the members of the other sex
- A Matching is a one-one correspondence between **M** and **W**
- Goal: Find a matching that is "**stable**", if it exists

# Stable Marriage(SM) problem

- Two disjoint sets **M**(men) and **W**(women), each of size **n**
- Each person has a <u>strictly ordered</u> preference list of **all** the members of the other sex
- A Matching is a one-one correspondence between **M** and **W**
- Goal: Find a matching that is “**stable**”, if it exists
- Fact: Stable matchings always exist for any SM instance

  Proof: Algorithmic !

# Stable Marriage(SM) problem

- Two disjoint sets **M**(men) and **W**(women), each of size **n**
- Each person has a strictly ordered preference list of **all** the members of the other sex
- A Matching is a one-one correspondence between **M** and **W**
- ~~Goal: Find a matching that is “**stable**”, if it exists~~
- Fact: Stable matchings always exist for any SM instance

  Proof: Algorithmic !
- Goal: Find a matching that is “**stable**”

# Matching-partners

- If **m** and **w** are matched(partners) in some matching **M**:

  **m** = $p_M$(**w**) = **M**-partner of **w**

  **w** = $p_M$(**m**) = **M**-partner of **m**

# Blocking Pair

(**m,w**) are said to be a **blocking pair** for a matching **M** if:

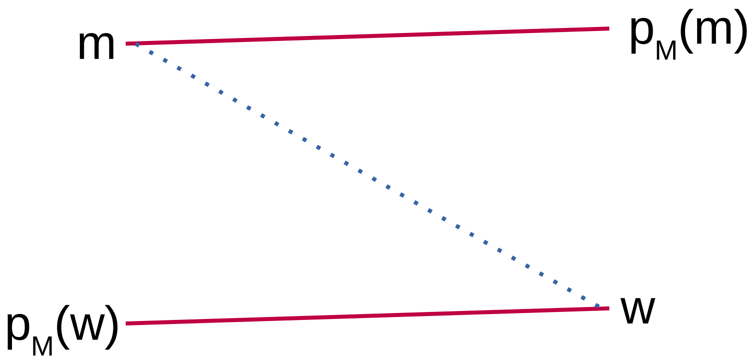
- **m** and **w** are not partners in **M**
- **m** prefers **w** to $p_M$(**m**)
- **w** prefers **m** to $p_M$(**w**)

# Blocking Pair

(**m,w**) are said to be a **blocking pair** for a matching **M** if:

- **m** and **w** are not partners in **M**
- **m** prefers **w** to $p_M$(**m**)
- **w** prefers **m** to $p_M$(**w**)

(m,w) is a blocking pair for M




m: .......w......$p_M$(m).......

w: .......m......$p_M$(w).......

# Stability

- A Matching is **unstable** if it has at least one blocking pair
- Absence of blocking pair <=> Matching is **stable**

# Gale-Shapley algorithm

Basic idea:

- Men propose, Women accept/reject
- Repeat until all men(and women) are engaged

# Gale-Shapley algorithm

Assign each person to be free

While some man **m** is free:

  **w** = first woman on **m**’s list to whom **m** hasn’t proposed yet

  if **w** is free:

    assign **m** and **w** as engaged [to each other]

  else

    if **w** prefers **m** to her fiance **m'**:

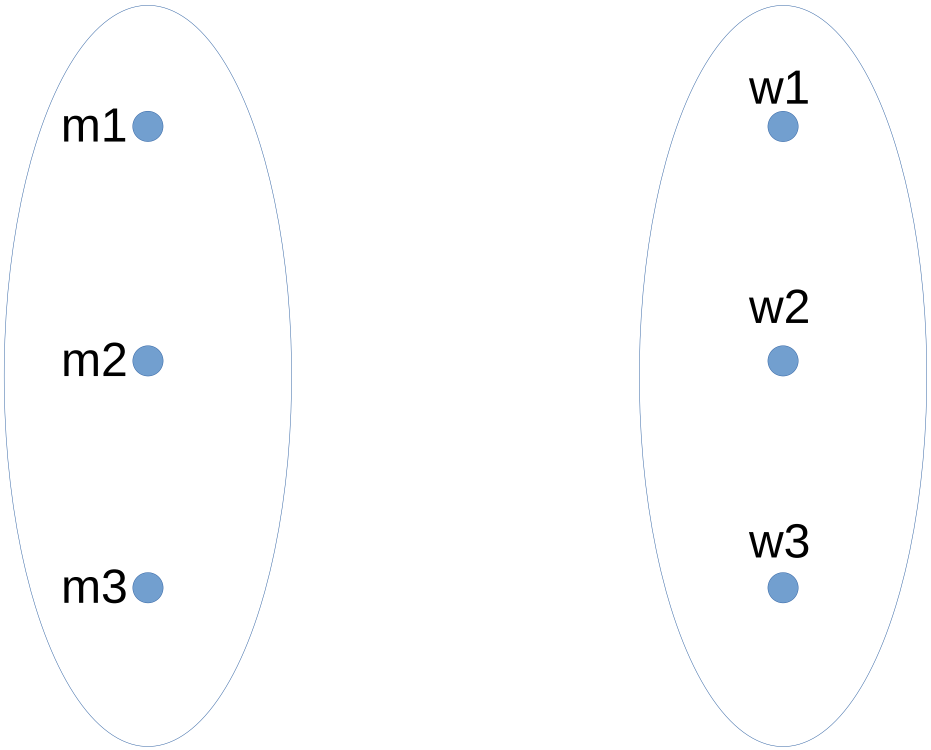
      assign **m** and **w** as engaged and **m'** as free

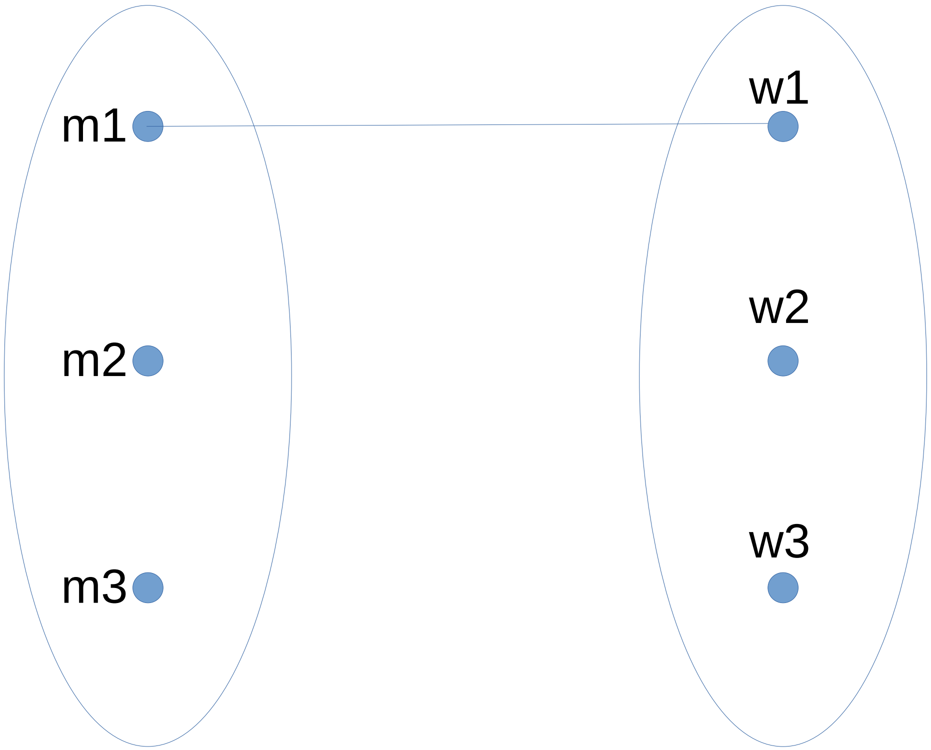
    else

      **w** rejects **m** [**m** remains free]

m1: w1 > w3 > w2
m2: w3 > w2 > w1
m3: w1 > w2 > w3

w1: m2 > m3 > m1
w2: m1 > m2 > m3
w3: m2 > m1 > m3

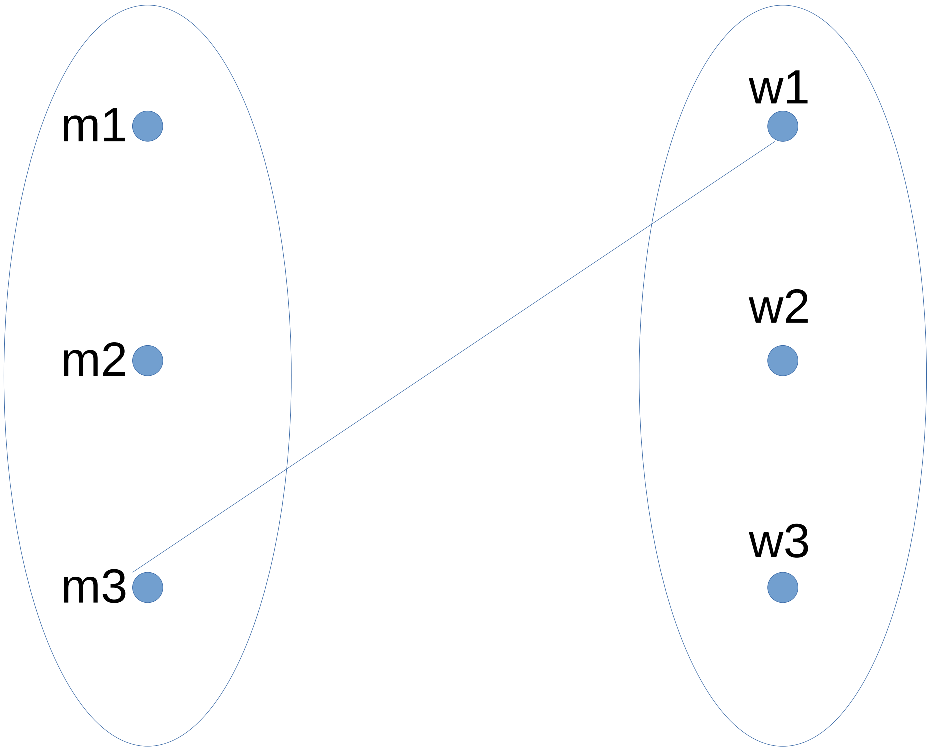





m1: w1 > w3 > w2
m2: w3 > w2 > w1
m3: w1 > w2 > w3

w1: m2 > m3 > m1
w2: m1 > m2 > m3
w3: m2 > m1 > m3

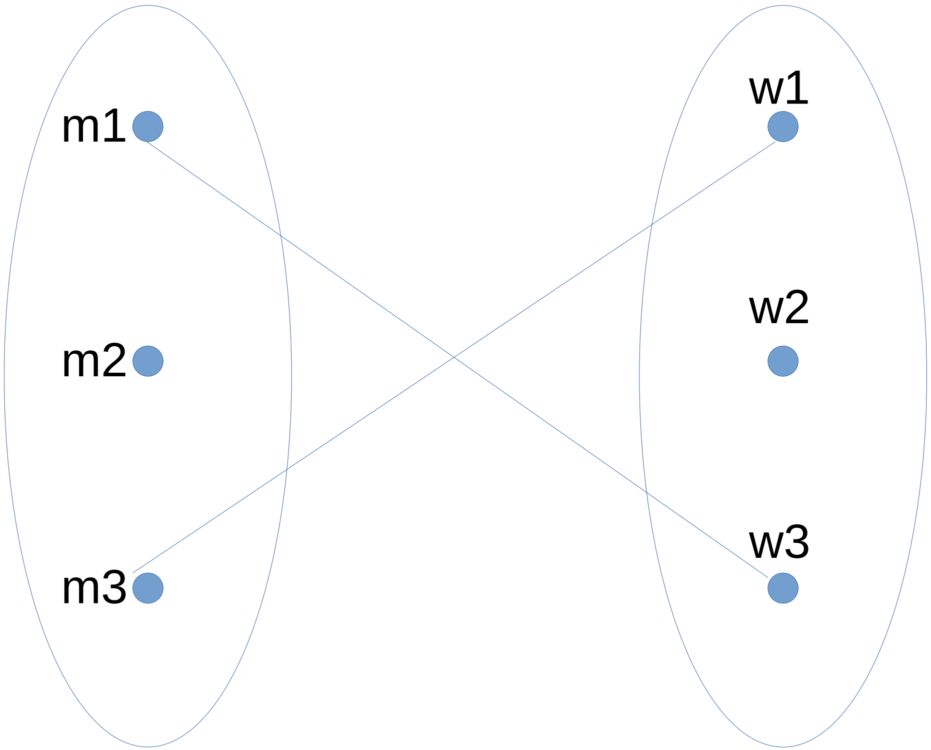
M1 proposes w1, w1 accepts m1




m1: w1 > w3 > w2
m2: w3 > w2 > w1
m3: w1 > w2 > w3

w1: m2 > m3 > m1
w2: m1 > m2 > m3
w3: m2 > m1 > m3

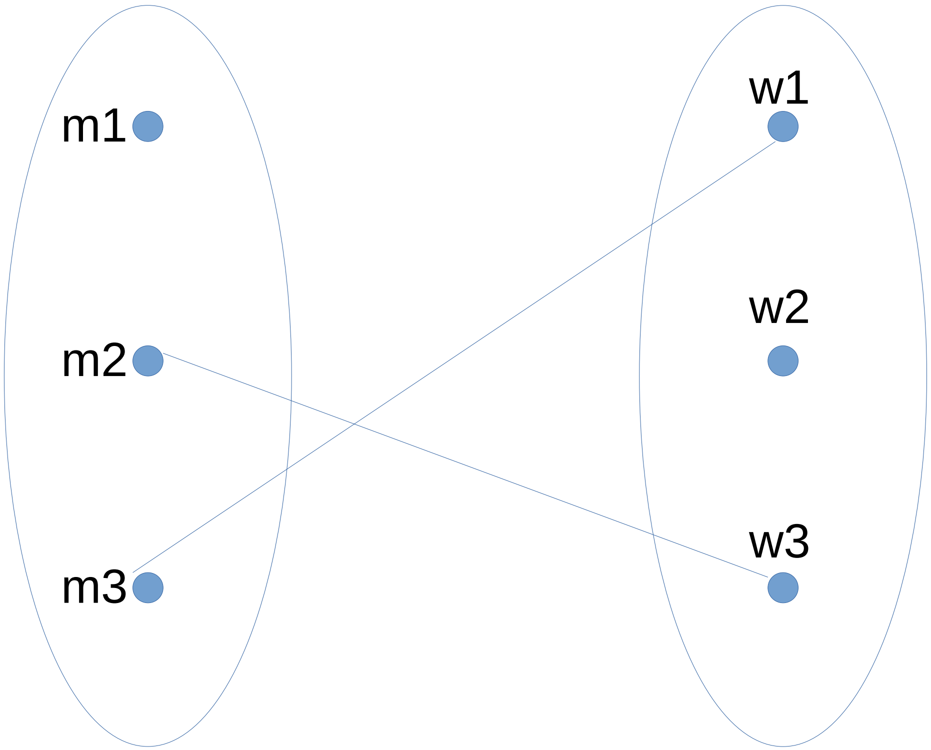
M1 proposes w1, w1 accepts m1
M3 proposes w1, w1 accepts m3, m1 free




m1: w1 > w3 > w2
m2: w3 > w2 > w1
m3: w1 > w2 > w3

w1: m2 > m3 > m1
w2: m1 > m2 > m3
w3: m2 > m1 > m3

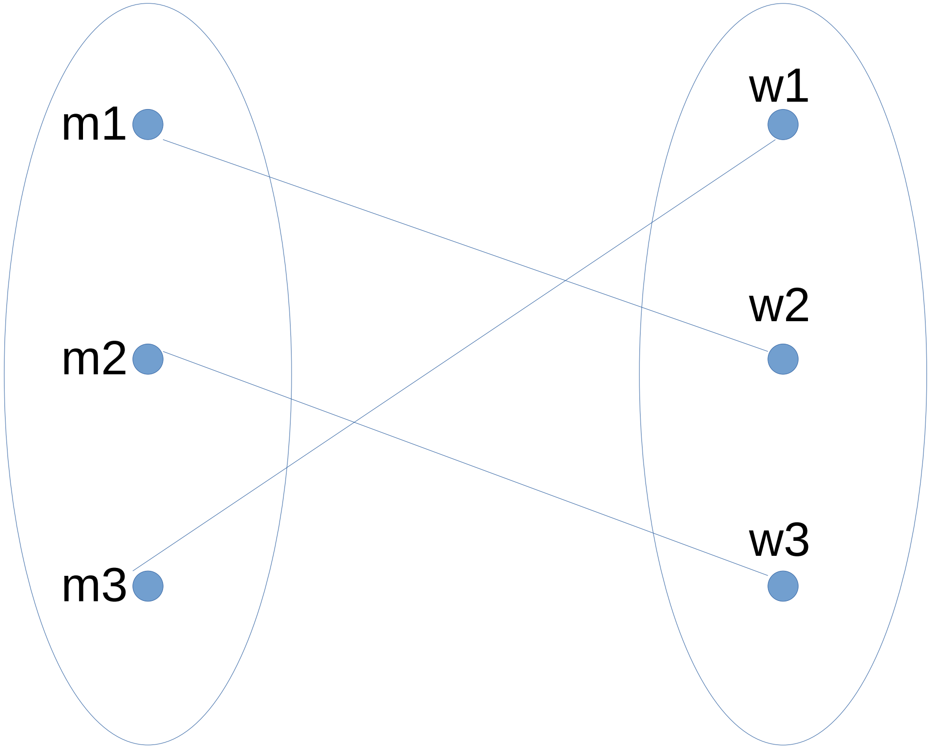
M1 proposes w1, w1 accepts m1
M3 proposes w1, w1 accepts m3, m1 free
M1 proposes w3, w3 accepts m1




m1: w1 > w3 > w2
m2: w3 > w2 > w1
m3: w1 > w2 > w3

w1: m2 > m3 > m1
w2: m1 > m2 > m3
w3: m2 > m1 > m3

M1 proposes w1, w1 accepts m1
M3 proposes w1, w1 accepts m3, m1 free
M1 proposes w3, w3 accepts m1
M2 proposes w3, w3 accepts m2, m1 free




m1:  w1 > w3 > w2
m2:  w3 > w2 > w1
m3:  w1 > w2 > w3

w1:  m2 > m3 > m1
w2:  m1 > m2 > m3
w3:  m2 > m1 > m3

M1 proposes w1, w1 accepts m1
M3 proposes w1, w1 accepts m3, m1 free
M1 proposes w3, w3 accepts m1
M2 proposes w3, w3 accepts m2, m1 free
M1 proposes w2, w2 accepts m1

# Proof of termination

Suffices to show that no man can be rejected by all women.

**Assume that a man has been rejected by all women.**

A woman can reject only when she is engaged.
Once a woman is engaged she never becomes free again.

Therefore, rejection of a man by the last woman on his list =>  All the women were already engaged. But, #men = #women = n, and no has more than one fiance, a contradiction.

# Proof of stability

- Let **M** be the matching generated by the algorithm

  **Assume ∃ a blocking pair (m,w)**

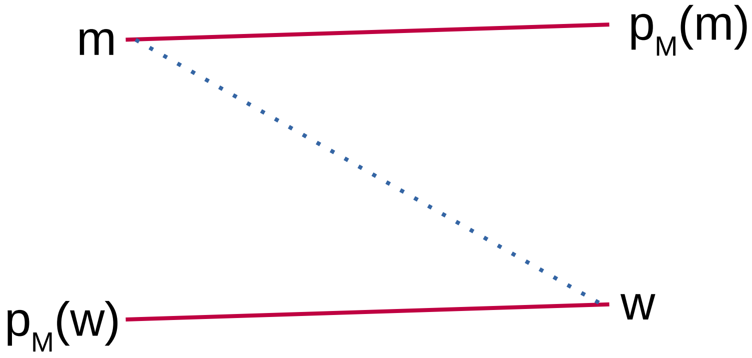
- **m** must have proposed **w** and got rejected
- **w** rejected **m**,

  since it must have had a better partner,

  say **m'** and thus would never accept $\mathbf{p_M(w)}$,

  a contradiction !

(m,w) is a blocking pair for M




m: .............w......$p_M(m)$.......
w: ......m'....m......$p_M(w)$.......

All possible executions of GS algorithm yield the same stable matching M, and in M, each man gets his best stable partner that he can have in any stable matching

**Proof**

- Let E be an arbitrary excution of the algorithm, that gives stable matching M
- Assume M' is another stable matching s.t:
- m prefers w' = $p_{M'}$(m) to w = $p_M$(m).
- during execution E, w' must have rejected m.
- Suppose, w.l.o.g, this is the first time during E that a woman rejected a stable partner.
- Let m' be the man w' was engaged to (for whom she rejected m)
- m' can have no stable partner whom he prefers to w' (since no woman had previously rejected a stable partner)
- Thus m' prefers w' to $p_{M'}$(m') and (m', w') blocks M', a contradiction !

All possible executions of GS algorithm yield the same stable matching M, and in M, each man gets his best stable partner that he can have in any stable matching

**Proof**

- Let E be an arbitrary excution of the algorithm, that gives stable matching M
- Assume M' is another stable matching s.t:
- m prefers w' = $p_{M'}(m)$ to w = $p_M(m)$.
- during execution E, w' must have rejected m.
- Suppose, w.l.o.g, this is the first time during E that a woman rejected a stable partner.
- Let m' be the man w' was engaged to (for whom she rejected m)
- m' can have no stable partner whom he prefers to w' (since no woman had previously rejected a stable partner)
- Thus m' prefers w' to $p_{M'}(m')$ and (m', w') blocks M', a contradiction !

Man-optimal stable matching !

All possible executions of GS algorithm yield the same stable matching M, and in M, each man gets his best stable partner that he can have in any stable matching

**Proof**

- Let E be an arbitrary excution of the algorithm, that gives stable matching M
- Assume M' is another stable matching s.t:
- m prefers w' = $p_{M'}(m)$ to w = $p_M(m)$.
- during execution E, w' must have rejected m.
- Suppose, w.l.o.g, this is the first time during E that a woman rejected a stable partner.
- Let m' be the man w' was engaged to (for whom she rejected m)
- m' can have no stable partner whom he prefers to w' (since no woman had previously rejected a stable partner)
- Thus m' prefers w' to $p_{M'}(m')$ and (m', w') blocks M', a contradiction !

Man-optimal stable matching !
Also woman-pessimal !

# Structure of Stable Matchings

- Let S = set of all possible Stable Matchings of an SM instance

  Dominance relation(≤):

    M dominates M' (M≤M') if every man has atleast as good a partner in M  as he has in M'

- (S, ≤) forms a lattice
- Man-optimal stable matching and woman-optimal stable matching are the greatest and least elements.

# Some common Extensions

- Sets of unequal size
- Unacceptable partners/Incomplete preference lists
- Indifference

# Some common Extensions

- Sets of unequal size
- Unacceptable partners/Incomplete preference lists
- Indifference

Definition of “stability” changes for each variant !

# Hospitals-Residents(HR) problem

- An <u>asymmetric</u> extension of the SM problem
- Consists of two sets:

  **H**: set of hospitals

  **R**: set of residents
- Each resident can be attached to one hospital
- Each hospital can be attached to *upto* a <u>fixed number(capacity)</u> of residents

- A Matching in the **HR** problem is:

  a (partial) mapping from R to H such that no of residents assigned to a hospital does not exceed the hospital’s capacity.

# Stability

Matching **M** is **unstable** if ∃ (r,h), **r** ϵ **R**, **h** ϵ **H** such that:

- **h** and **r** are acceptable to each other
- Either **r** is unmatched **OR r** prefers **h** to its assigned hospital
- Either **h** doesn't have all its places filled in **M OR h** prefers **r** to atleast one of its residents

# Similarity of HR with SM

- Generalized GS algorithms(hospital-oriented and resident-oriented)
- All possible executions of the algorithms terminate with the same result
- Hospital-optimal and resident-optimal matchings for the corresponing algorithms
- Forms a lattice with a similar dominance relation as in SM

# Hospital oriented GS algorithm

- Assign each **r** ϵ **R** to be free
- Assign each **h** ϵ **H** to be totally unsubscribed
- While [some **h** ϵ **H** is undersubscribed AND **h**’s list contains an **r** ϵ **R** not provisionally assigned to **h**]:

    **r** = first such resident on **h**’s list

    if **r** is already assigned, say to **h'** :

        break the provisional assignment of **r** to **h'**

    provisionally assign **r** to **h**

    for each successor **h'** of **h** on **r**’s list:

        remove **h'** and **r** from each other’s list

# Hospital oriented GS algorithm

- Assign each **r** ϵ **R** to be free
- Assign each **h** ϵ **H** to be totally unsubscribed
- While [some **h** ϵ **H** is undersubscribed AND **h**’s list contains an **r** ϵ **R** not provisionally assigned to **h**]:

    **r** = first such resident on **h**’s list

    if **r** is already assigned, say to **h'** :

        break the provisional assignment of **r** to **h'**

    provisionally assign **r** to **h**

    for each successor **h'** of **h** on **r**’s list:

        remove **h'** and **r** from each other’s list

On termination, it generates a hospital-optimal and resident-pessimal matching

# Other Applications

- HR with lower quotas
- Student-Project Allocation

# References


- Dan Gusfield and Robert W. Irving. 1989. The stable marriage problem: structure and algorithms. MIT Press, Cambridge, MA, USA.
- Algorithmics of Matching Under Preferences: David F Manlove
- Gale, David, and Lloyd S. Shapley. "College admissions and the stability of marriage." The American Mathematical Monthly 69.1 (1962): 9-15.
- Roth, Alvin E. "The evolution of the labor market for medical interns and residents: a case study in game theory." Journal of political Economy 92.6 (1984): 991-1016.
- Stable Marriage and Its Relation to Other Combinatorial Problems: An Introduction to the Mathematical Analysis of Algorithms: Knuth, D.E, American Mathematical Soc

# Other Useful Resources

- https://www.youtube.com/watch?v=pvPtfdE4NSc
- https://www.youtube.com/watch?v=VpDEuindebE
- https://github.com/severus-tux/StableMatching

Thank You !