CPE231RC — Algorithms — Lecture9

Iterative Improvement

Dr. Prapong Prechaprapranwong

Computer Engineering

King Mongkut's University of Technology Thonburi

Iterative Improvement

Iterative Improvement or **Iterative Refinement** – constructs a solution to an optimization problem by

1) start with some feasible solution
2) repeat to improve a value of objective function by a small step until no change improves the value objective function
3) stop and return the last feasible solution as an optimal solution

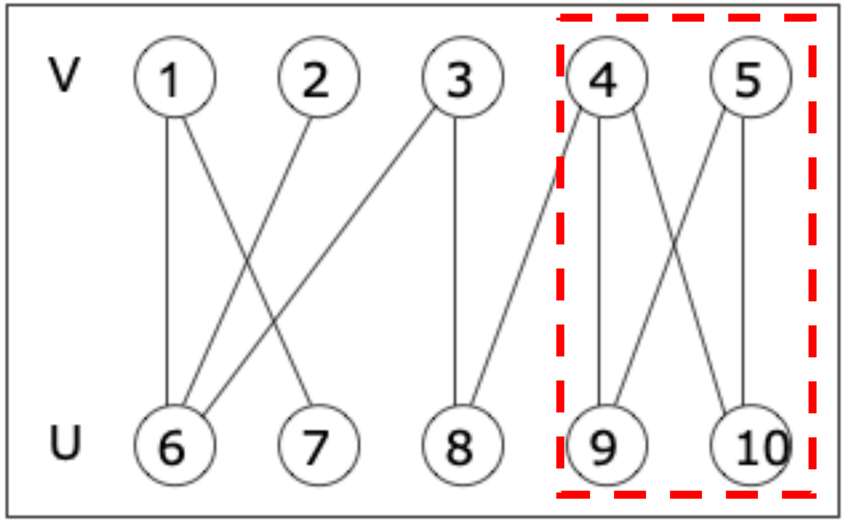
Examples:

- Maximum matching in Bipartite graphs
- Stable marriage problem

Maximum matching in Bipartite graphs problem

Bipartite Graph

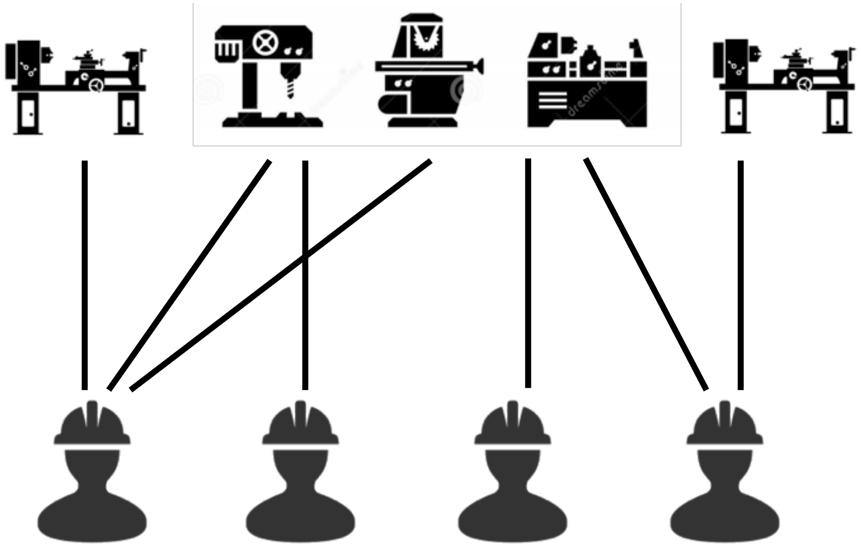
- A graph whose vertices can be partitioned into two disjoint sets *V* and *U*, not necessarily of the same size, so that every edge connects a vertex in *V* to a vertex in *U*
- A graph is bipartite if and only if <u>it does not have a cycle of an odd length</u>.



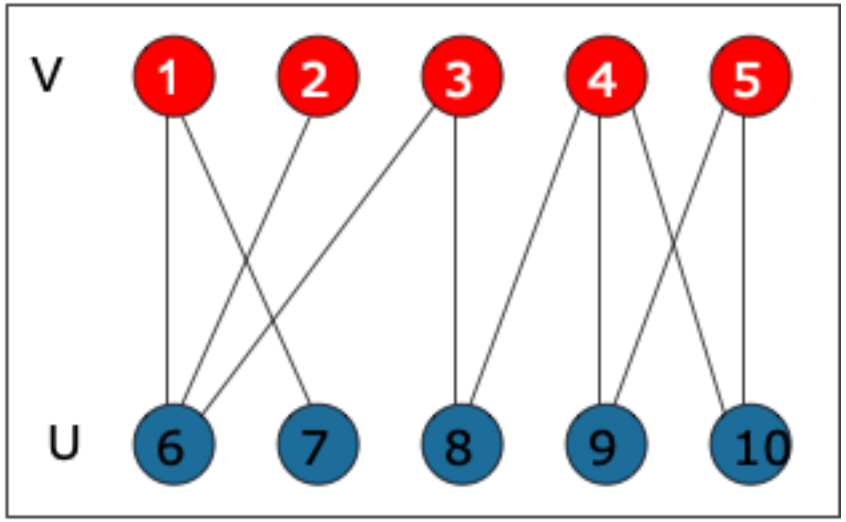
cycle with <u>even</u> length
so still being a bipartite

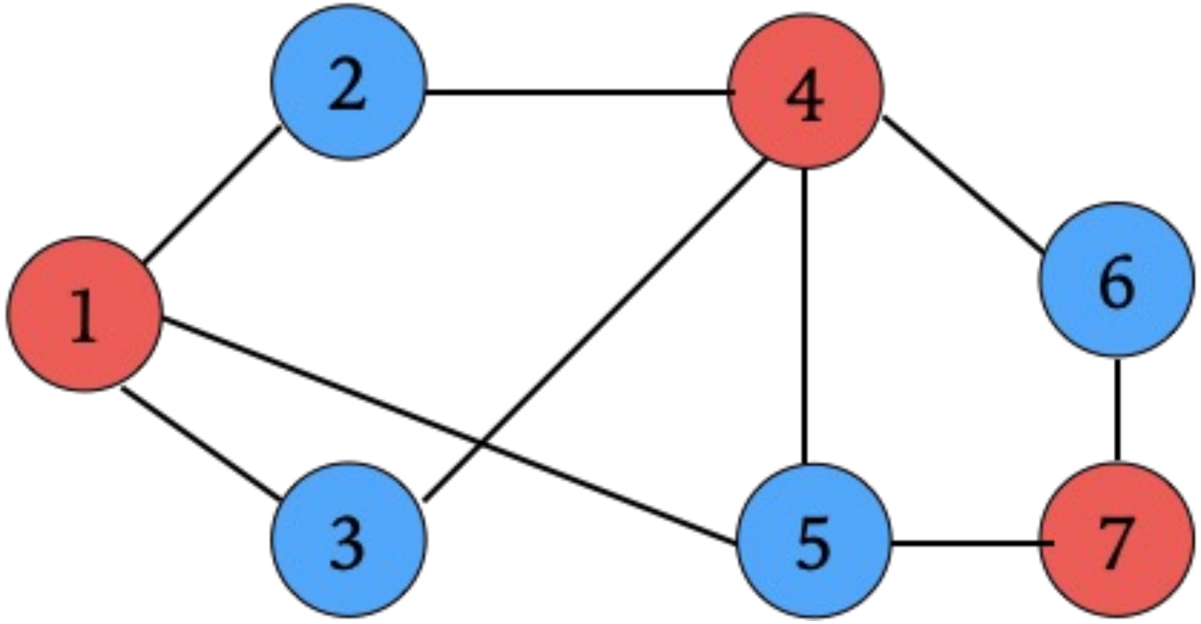
Machines

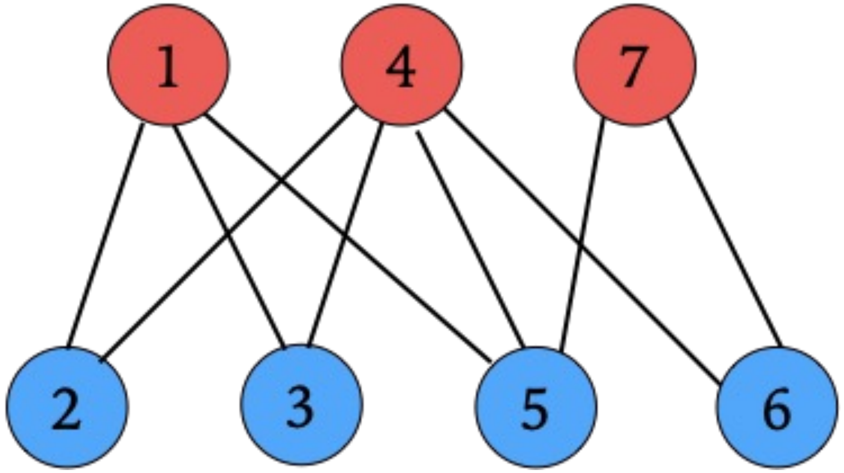
Workers



A bipartite graph is 2-colorable: the vertices can be colored in two colors so that every edge has its vertices colored differently





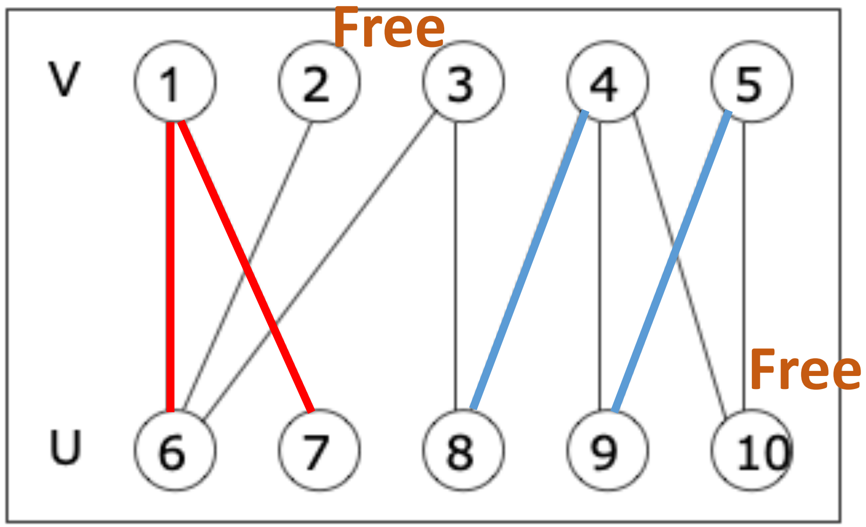


no two adjacent vertices have the same color

Matching in a Bipartite Graph

- **matching** - a subset of edges where no two edges share a vertex

{(1,6),(1,7)}
not a matching



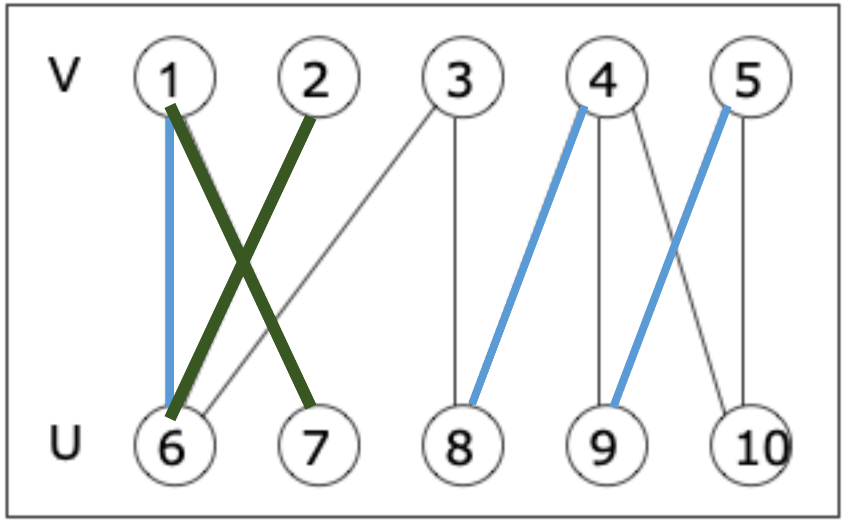
{(4,8),(5,9)}
matching

- **maximum matching** is a matching with the largest number of edges
  - always exists in Bipartite graph but not always unique

- For a given matching M, a vertex is called ***free*** (or unmatched) if it is not an endpoint of any edge in M; otherwise, a vertex is said to be ***matched***
- If every vertex is matched, then M is a **maximum matching**
- If there are unmatched or free vertices, then M may be able to be improved by increasing a matching - adding an edge connecting two free vertices e.g., (1,6)

Augmenting Path

Augmenting path for a matching M is a path from <u>a free vertex in V to a free vertex in U</u> whose edges alternate between edges in M^c and edges in M, and the last edge is in M^c



M = {(1,6), (4,8), (5,9)}

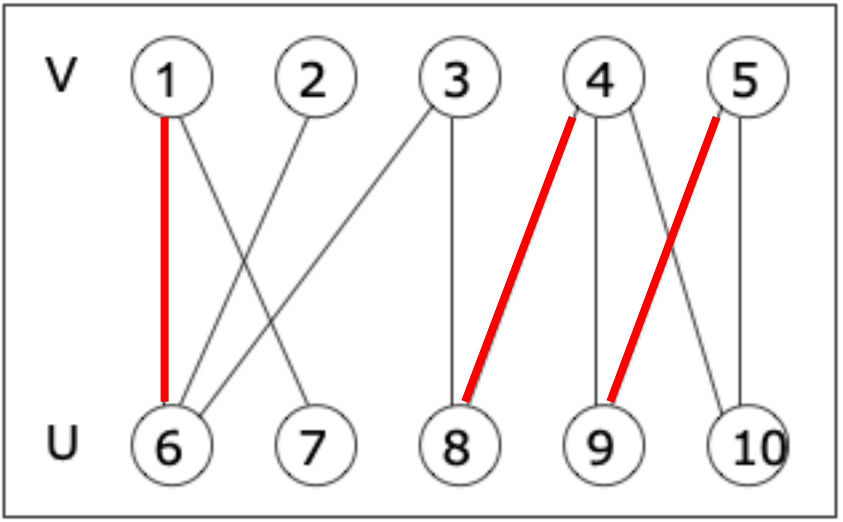
M^c = {(1,7),(2,6),(3,6),(3,8),(4,9),(4,10),(5,10)}

The length of an augmenting path is always odd.

Augmentation along path {(2,6),(6,1),(1,7)}

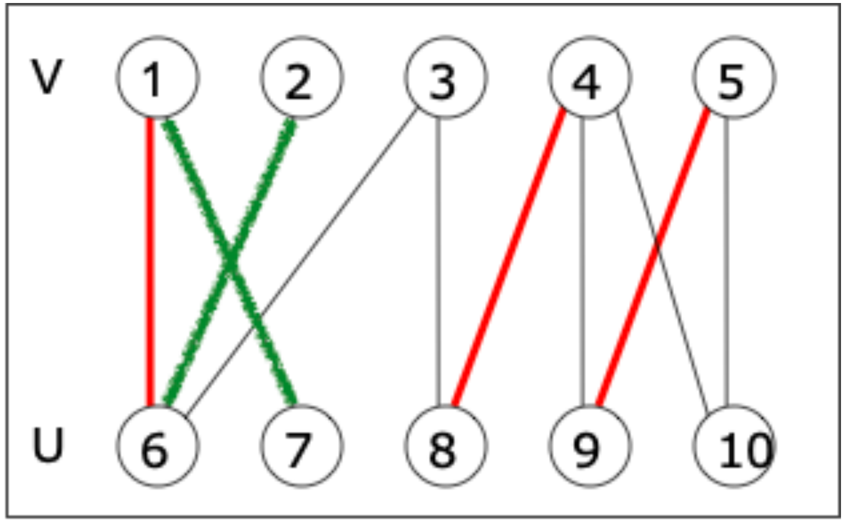
One-edge path between two free vertices is special case of augmenting path

Example:



M = {(1,6), (4,8), (5,9)}

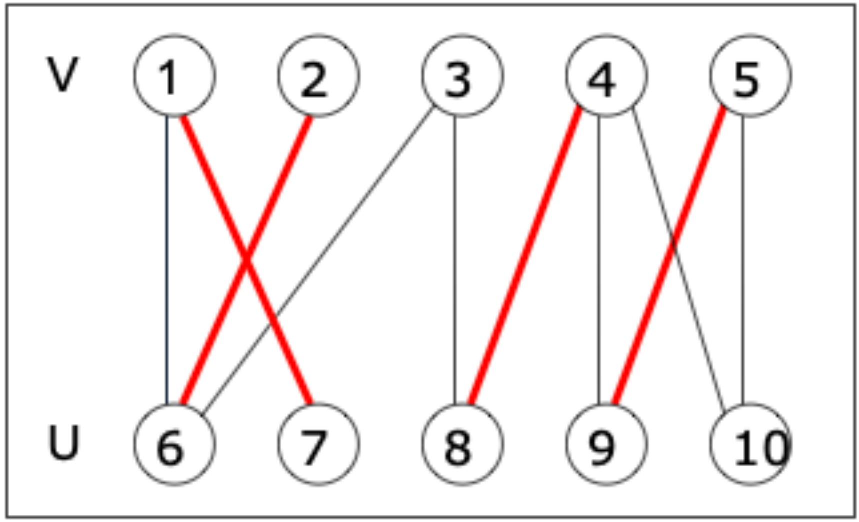
M^c = {(1,7), (2,6), (3,6),(3,8), (4,9), (4,10), (5,10)}



Augmentation along path {(2,6),(6,1),(1,7)}

M = {~~(1,6)~~, (4,8), (5,9)}

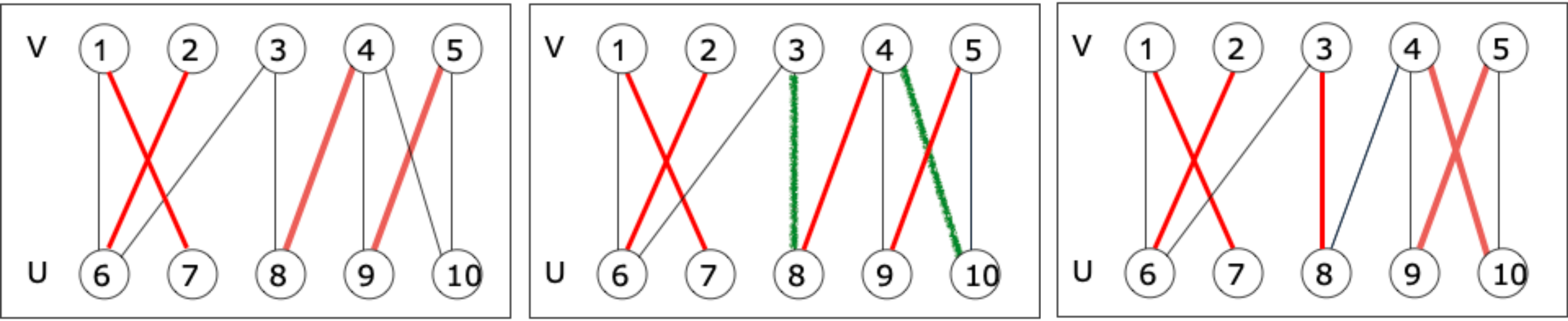
M^c = {~~(1,7)~~, ~~(2,6)~~, (3,6), (3,8), (4,9), (4,10), (5,10)}



Add odd-numbered edges (2,6), (1,7) to M

M = {(4,8), (5,9),(2,6),(1,7)}

M^c = {(3,6), (3,8), (4,9), (4,10), (5,10),(1,6)}



M = {(4,8), (5,9),(2,6),(1,7)}

M^c = {(3,6), (3,8), (4,9), (4,10), (5,10),(1,6)}

Augmentation along path {(3,8),(8,4),(4,10)}

M = {~~(4,8)~~, (5,9),(2,6),(1,7)}

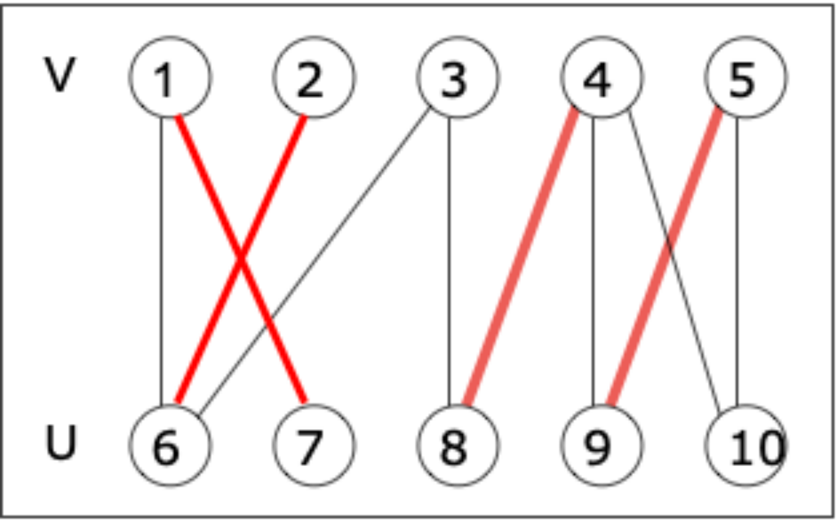
M^c = {(3,6), ~~(3,8)~~, (4,9), ~~(4,10)~~, (5,10),(1,6)}

Add odd-numbered edges (3,8),(4,10)

M = {(5,9),(2,6),(1,7),(3,8),(4,10)}

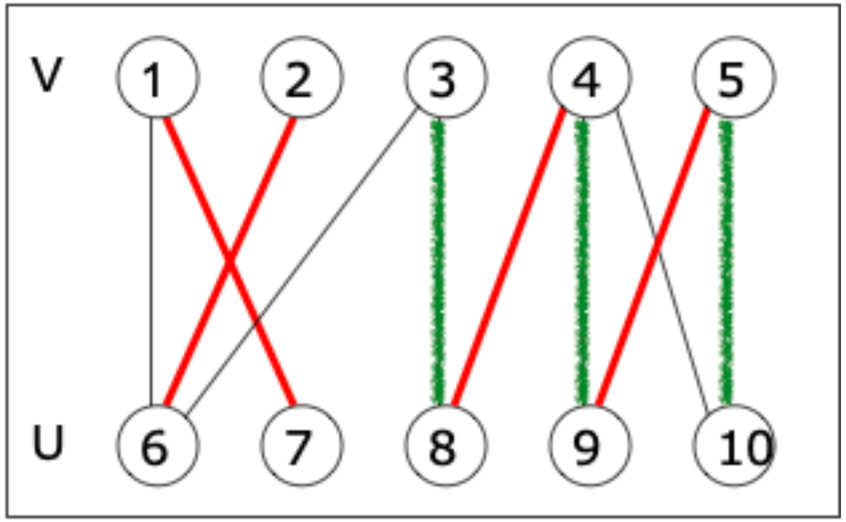
M^c = {(3,6),(4,9), (5,10),(1,6), (4,8)}

A matching M is maximum if and only if there exists <u>no augmenting path</u> with respect to M



M = {(4,8), (5,9),(2,6),(1,7)}

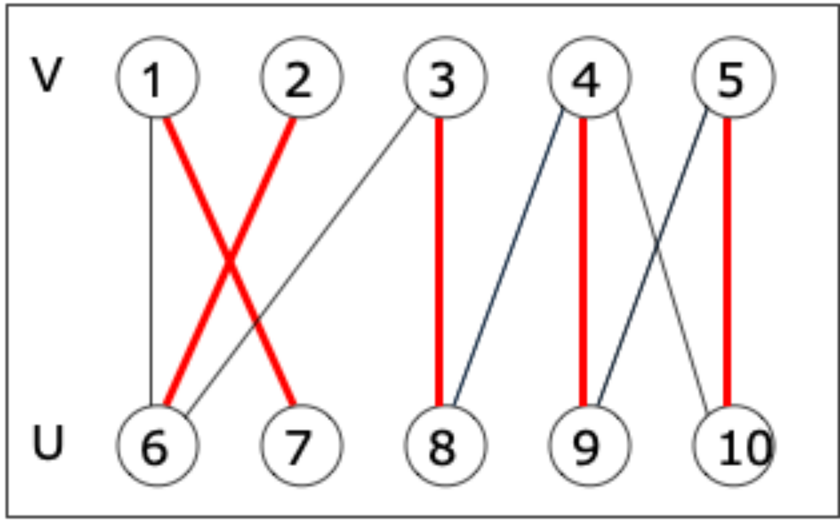
M^c = {(3,6), (3,8), (4,9), (4,10), (5,10),(1,6)}



Augmentation along path {(3,8),(8,4),(4,9),(9,5),(5,10)}

M = {~~(4,8)~~, ~~(5,9)~~,(2,6),(1,7)}

M^c = {(3,6), ~~(3,8)~~, ~~(4,9)~~, (4,10), ~~(5,10)~~, (1,6)}



Add odd-numbered edges (3,8),(4,9),(5,10)

M = {(2,6),(1,7),(3,8),(4,9),(5,10)}

M^c = {(3,6), (4,10),(1,6),(4,8), (5,9)}

Marriage Matching Problem

- Set Y = $\{m_1, \ldots, m_n\}$ of n men and set X = $\{w_1, \ldots, w_n\}$ of n women.
- Each man has a ranking list of the women, and each woman has a ranking list of the men (with no ties in these lists).

men's preferences

	1st	2nd	3rd
Bob	Lea	Ann	Sue
Jim	Lea	Sue	Ann
Tom	Sue	Lea	Ann

women's preferences

	1st	2nd	3rd
Ann	Jim	Tom	Bob
Lea	Tom	Bob	Jim
Sue	Jim	Tom	Bob

- A marriage matching M is a set of n pairs {(mi, wj)}.
- Can be represented by a bipartite graph.

Ex: M = {(Bob, Lea), (Jim, Sue), (Tom, Ann)}

M = {(Bob, Ann), (Jim, Sue), (Tom, Lea)}

- Total # possible matchings is n!

women's preferences

	1st	2nd	3rd
Ann	Jim	Tom	Bob
Lea	Tom	Bob	Jim
Sue	Jim	Tom	Bob

men's preferences

	1st	2nd	3rd
Bob	Lea	Ann	Sue
Jim	Lea	Sue	Ann
Tom	Sue	Lea	Ann

	Ann	Lea	Sue
Bob	2,3	1,2	3,3
Jim	3,1	1,3	2,1
Tom	3,2	2,1	1,2

Ranking matrix

- A pair (m, w) is said to be a **blocking pair** for matching **M** if man *m* and woman *w* are not matched in **M** but prefer each other to their mates in **M**.

- Consider M = {(Bob, Ann), (Jim, Lea), (Tom, Sue)}
  - Bob prefers Lea to Ann, Lea prefers Bob to Jim
  - (Bob, Lea) is a blocking pair for M.

	Ann	Lea	Sue
Bob	2,3	1,2	3,3
Jim	3,1	$\boxed{1,3}$	2,1
Tom	3,2	2,1	$\boxed{1,2}$

For each pair (m,w) in M,
– if w is not the 1st in his list, get the preferred mate w'
– check if w' is mated in another pair in M. If w' prefers m to her mate m', we have blocking pair (m,w')

Jim and Tom already have their most preferred mate.
but Bob?

Stable Marriage Problem

find a stable marriage matching for men's and women's given preferences.

- A marriage matching M is called **stable** if there is no **blocking pair** for it; otherwise, it's called **unstable**.

Which one below is stable marriage?

M1 = {(Bob, Ann),(Jim, Lea),(Tom, Sue)}

	Ann	Lea	Sue
Bob	$\boxed{2,3}$	1,2	3,3
Jim	3,1	$\boxed{1,3}$	2,1
Tom	3,2	2,1	$\boxed{1,2}$

M2 = {(Bob, Ann),(Jim, Sue),(Tom, Lea)}

	Ann	Lea	Sue
Bob	$\boxed{2,3}$	1,2	3,3
Jim	3,1	1,3	$\boxed{2,1}$
Tom	3,2	$\boxed{2,1}$	1,2

Stable Marriage Algorithm

Algorithm guarantees that
•Everyone gets married
•The marriages are stable

Step 0: Start with all the men and women being free

Step 1: While there are free men, arbitrarily select one of them (m) and do the following:

- **Proposal** *m* proposes to *w*, the next woman on his preference list (highest-rank not reject him before)
- **Response** If *w* is free, she accepts the proposal to be matched with *m*. If she is not free, she compares *m* with her current mate. If she prefers *m* to him, she accepts *m*'s proposal, making her former mate free; otherwise, she simply rejects *m*'s proposal, leaving *m* free

Step 2: Return the set of n matched pairs

Example:

Free men:
Bob, Jim, Tom

	Ann	Lea	Sue
Bob	2, 3	$\boxed{1,2}$	3, 3
Jim	3, 1	1, 3	2, 1
Tom	3, 2	2, 1	1, 2

Bob proposed to Lea
Lea accepted

Free men:
Jim, Tom

	Ann	Lea	Sue
Bob	2, 3	$\boxed{1,2}$	3, 3
Jim	3, 1	1, 3	2, 1
Tom	3, 2	2, 1	1, 2

Jim proposed to Lea
Lea rejected

Free men:
Jim, Tom

	Ann	Lea	Sue
Bob	2, 3	$\boxed{1,2}$	3, 3
Jim	3, 1	1, 3	$\boxed{2,1}$
Tom	3, 2	2, 1	1, 2

Jim proposed to Sue
Sue accepted

Free men:
Tom

	Ann	Lea	Sue
Bob	2, 3	$\boxed{1,2}$	3, 3
Jim	3, 1	1, 3	$\boxed{2,1}$
Tom	3, 2	2, 1	$\underline{1, 2}$

Tom proposed to Sue
Sue rejected

Free men:
Tom

	Ann	Lea	Sue
Bob	2, 3	1, 2	3, 3
Jim	3, 1	1, 3	$\boxed{2,1}$
Tom	3, 2	$\boxed{2,1}$	1, 2

Tom proposed to Lea
Lea replaced Bob with Tom

Free men:
Bob

	Ann	Lea	Sue
Bob	$\boxed{2,3}$	1, 2	3, 3
Jim	3, 1	1, 3	$\boxed{2,1}$
Tom	3, 2	$\boxed{2,1}$	1, 2

Bob proposed to Ann
Ann accepted

M = {(Bob,Ann), (Jim,Sue), (Tom,Lea)}

Stable Marriage Algorithm

- The algorithm terminates after no more than n^2 iterations with a stable marriage output
- Matching produced always **man-optimal**:
- Each man gets the highest-ranked woman possible on his list under any stable marriage.
- One can obtain **the woman-optimal** matching by making women propose to men
- A man (woman) optimal matching is unique for a given set of participant preferences

Assignment: From the following instance of the stable marriage problem, find a stable marriage matching by applying the stable marriage algorithm in its <u>women proposing version.</u>