# CS 207

## Discrete Structures

Nutan Limaye

24 SEP 2013

## Last Class

1. $2^{(l-2)/2} \leq R(l,l) \leq 2^{2(l-1)} / \sqrt{2(l-1)}$

2. Started with Module - 3 (graph theory)

A graph $G = (V, E)$ is called bipartite if the vertices of the graph can be partitioned $V = X \dot{\cup} Y$ s.t.

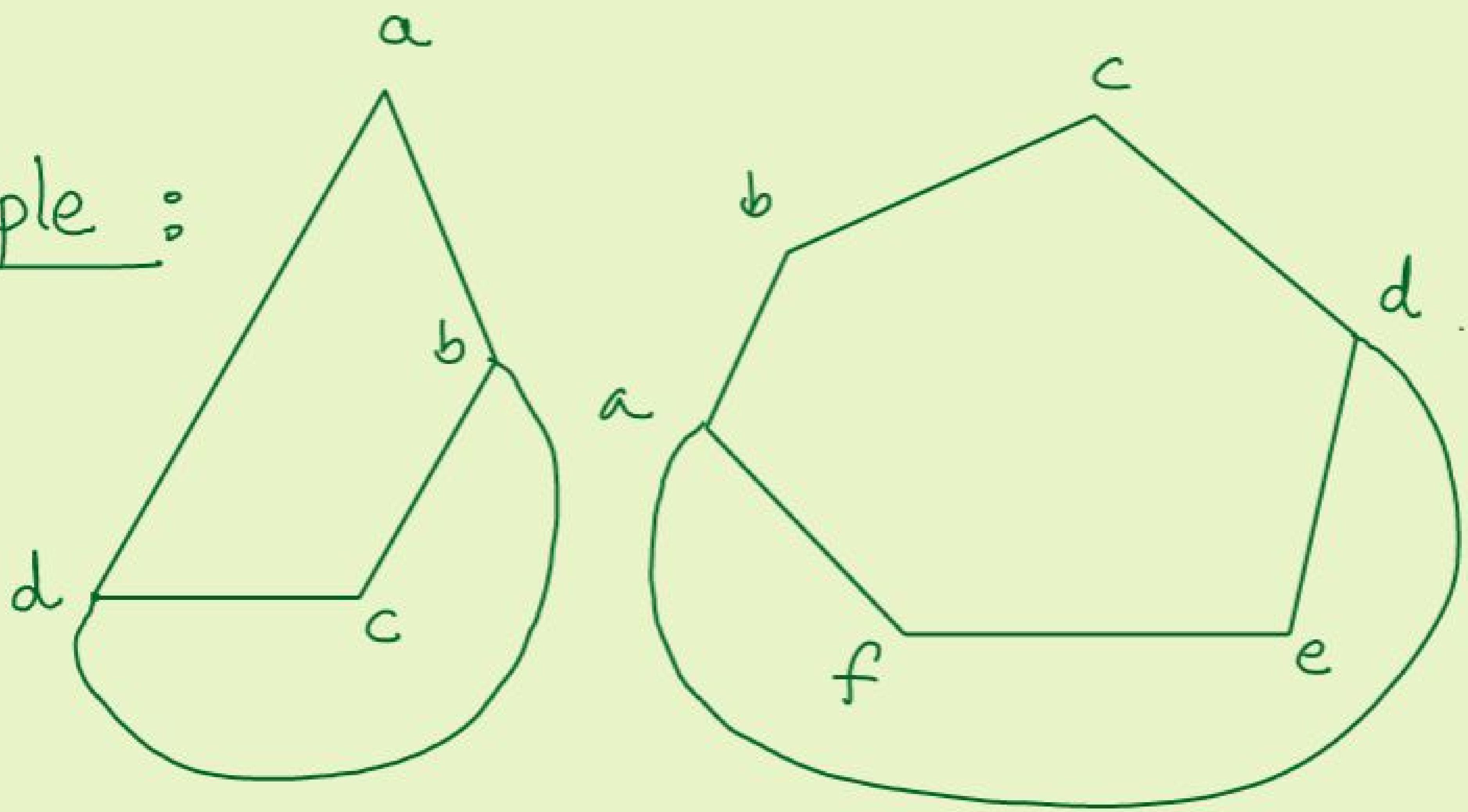
$\forall e = (u, v) \in E$ : either $u \in X$ and $v \in Y$

or $u \in Y$ and $v \in X$.

A graph $G = (V, E)$ is called bipartite if the vertices of the graph can be partitioned $V = X \dot{\cup} Y$ s.t.

$\forall\, e = (u, v) \in E$ : either $u \in X$ and $v \in Y$
or $u \in Y$ and $v \in X$.

Example :



## Some definitions: Let $G=(V,E)$ be a graph:

**Walk**: is a sequence of vertices $v_1, v_2, v_3 \dots v_k$ such that $\forall i \in [k-1]$, $(v_i, v_{i+1}) \in E$.

**Path**: a walk is a path if no vertex repeats it self.

**Closed walk**: a walk is called a closed walk if it starts and ends at the same vertex.

**Cycle**: a closed walk $v_1, v_2, \dots, v_\ell, v_1$ is called a cycle if $v_1 \neq v_2 \neq \dots \neq v_\ell$.

# Some more definitions

- length of a path $\triangleq$ # edges on the path.
- length of a cycle $\triangleq$ # edges on the cycle.
- a path (cycle) is called odd (even) if its length is odd (respectively, even).

__Theorem__: A graph is bipartite if and only if it does not have an odd cycle.

Theorem : A graph is bipartite if and only if it does not have an odd cycle.

proof : ($\Rightarrow$) First we will prove that if the graph is bipartite then it does not have an odd cycle.

Theorem: A graph is bipartite if and only if it does not have an odd cycle.

proof: ($\Rightarrow$) a cycle must begin & end in the same vertex.

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Longrightarrow$) a cycle must begin & end in the same vertex.

As $G = (V, E)$ is bipartite, let $V = X \dot{\cup} Y$ be its bipartition.

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Longrightarrow$) a cycle must begin & end in the same vertex, say $x \in X$.

$x = x_0, x_1, x_2, \ldots \ldots, x_{2k-1}, x_{2k}, x_{2k+1}$

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Longrightarrow$) a cycle must begin & end in the same vertex, say $x \in X$.

$$x = x_0, x_1, x_2, \ldots\ldots, x_{2k-1}, x_{2k}, x_{2k+1}$$

$$x \in X \longrightarrow \forall\, i \in [k+1],\ x_{2i} \in X,\ x_{2i-1} \in Y.$$

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: $(\Longrightarrow)$ a cycle must begin & end in the same vertex, say $x \in X$.

$$x = x_0, x_1, x_2, \ldots\ldots, x_{2k-1}, x_{2k}, x_{2k+1}$$

$$x \in X \longrightarrow \forall\, i \in [k+1],\ x_{2i} \in X,\ x_{2i-1} \in Y.$$

$$x \in X \longrightarrow x_{2k+1} \in Y \longrightarrow x \in Y$$

(a contradiction).

**Theorem:** A graph is bipartite if and only if it does not have an odd cycle.

**proof:** ($\Leftarrow$) If a graph does not have any odd cycle then it is bipartite.

**Theorem:** A graph is bipartite if and only if it does not have an odd cycle.

**proof:** ($\Leftarrow$) If every cycle has even length then the graph is bipartite.

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Leftarrow$)

- pick an arb vertex $v \in V$
- $\forall u \in V$, let $label(u) \leftarrow Dist(u, v)$

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Leftarrow$) ◦ pick an arb vertex $v \in V$

◦ $\forall u \in V$, let $label(u) \leftarrow Dist(u, v)$

$Dist(u, v)$ = length of the shortest path between $u$ & $v$.

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Leftarrow$)

- pick an arb vertex $v \in V$
- $\forall u \in V$, let $label(u) \leftarrow Dist(u,v)$

$V_{ODD} \leftarrow \{u \mid label(u) = 1 \bmod 2\}$

$V_{EVEN} \leftarrow \{u \mid label(u) = 0 \bmod 2\}$

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Leftarrow$)

- pick an arb vertex $v \in V$
- $\forall u \in V$, let $label(u) \leftarrow Dist(u, v)$

$V_{ODD} \leftarrow \{u \mid label(u) = 1 \bmod 2\}$

$V_{EVEN} \leftarrow \{u \mid label(u) = 0 \bmod 2\}$

If we prove that $V_{ODD}$ & $V_{EVEN}$ form a bipartition, than we will be done.

$\underline{\text{Claim}}$ : $\forall (x,y) \in E$ either $x \in V_{ODD}$ & $y \in V_{EVEN}$
or $x \in V_{EVEN}$ & $y \in V_{ODD}$.

$\underline{\text{proof}}$: Suppose $\exists (x,y)$ s.t. $x, y \in V_{ODD}$.

**Claim** : $\forall (x,y) \in E$ either $x \in V_{ODD}$ & $y \in V_{EVEN}$
or $x \in V_{EVEN}$ & $y \in V_{ODD}$.

**proof**: Suppose $\exists (x,y)$ s.t. $x, y \in V_{ODD}$.

$$\Downarrow$$

$\exists k \in \mathbb{N}$, $\exists$ path of length $2k+1$ from $v$ to $x$
& $\exists l \in \mathbb{N}$, $\exists$ path of length $2l+1$ from $v$ to $y$

**Claim** : $\forall (x,y) \in E$ either $x \in V_{ODD}$ & $y \in V_{EVEN}$

or $x \in V_{EVEN}$ & $y \in V_{ODD}$.

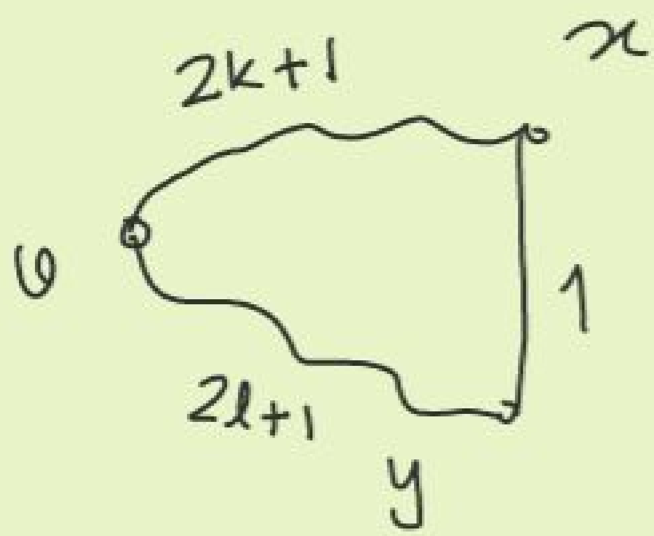
**proof** : Suppose $\exists (x,y)$ s.t. $x, y \in V_{ODD}$.

$$\Downarrow$$

$\exists k \in \mathbb{N}$, $\exists$ path of length $2k+1$ from $v$ to $x$

& $\exists l \in \mathbb{N}$, $\exists$ path of length $2l+1$ from $v$ to $y$

$$\Downarrow$$



**Claim:** $\forall (x,y) \in E$ either $x \in V_{ODD}$ & $y \in V_{EVEN}$

or $x \in V_{EVEN}$ & $y \in V_{ODD}$.

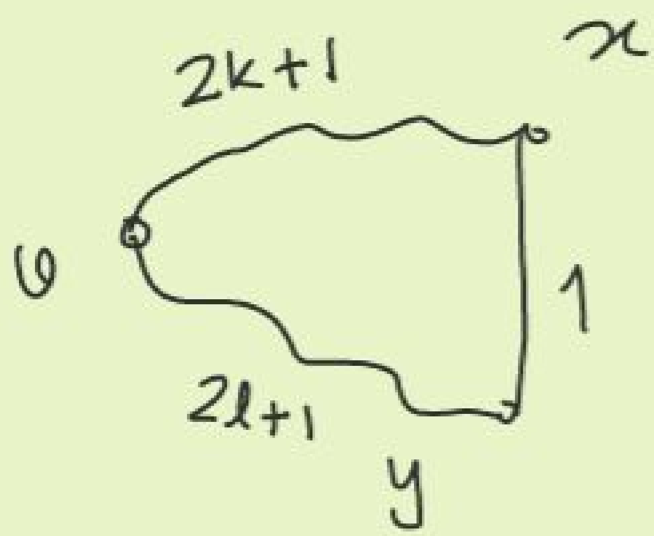
**proof:** Suppose $\exists (x,y)$ s.t. $x, y \in V_{ODD}$.

$\Downarrow$

$\exists k \in \mathbb{N}$, $\exists$ path of length $2k+1$ from $v$ to $x$

& $\exists l \in \mathbb{N}$, $\exists$ path of length $2l+1$ from $v$ to $y$

$\Downarrow$




$\therefore$ a cycle of length

$2(k+l+1)+1$

i.e. a contradition.

**Theorem**: A graph is bipartite if and only if it does not have an odd cycle.

**proof**: ($\Leftarrow$)

- pick an arb vertex $v \in V$
- $\forall u \in V$, let $label(u) \leftarrow Dist(u, v)$

$V_{ODD} \leftarrow \{u \mid label(u) = 1 \bmod 2\}$

$V_{EVEN} \leftarrow \{u \mid label(u) = 0 \bmod 2\}$

- $V_{ODD}$ and $V_{EVEN}$ form a bipartion of $V$

$\square$

**Lemma:** Every simple graph has a bipartite subgraph with $|E|/2$ edges.

**Lemma**: Every simple graph has a bipartite subgraph with $|E|/2$ edges.

Is the above statement true?

**Lemma**: Every simple graph has a bipartite subgraph with $\geq |E|/2$ edges.

Is the above statement true?

Does there exist a graph with exactly $\frac{|E|}{2}$ sized bipartite graph inside it?

# CS 207

# Discrete Structures

Nutan Limaye

26 SEP 2013

# Last Class

1. Bipartite graphs : a graph is bipartite iff it has no odd cycle.

2. Every simple graph has a bipartite subgraph of size $\geq \left\lceil \frac{|E|}{2} \right\rceil$

Today :

- Finish the proof of the statement presented last time
- Matchings — perfect matchings in bipartite graphs (Hall's condition).

Any simple graph has a bipartite subgraph with at least $\left\lceil \frac{|E|}{2} \right\rceil$ edges

Simple graph : A graph is said to be simple if

- $\leq 1$ edge between any pair of vertices
- no self-loops

Any simple graph has a bipartite subgraph with at least $\left\lceil \frac{|E|}{2} \right\rceil$ edges

Simple graph : A graph is said to be simple if

- $\leq 1$ edge between any pair of vertices
- no self-loops

Subgraph : $H = (V', E')$ is called a subgraph of $G = (V, E)$ if $E' \subseteq E$ and $V' = \{ v \mid \exists u \in V : (u, v) \in E' \}$

Any simple graph has a bipartite subgraph with at least $\left\lceil \frac{|E|}{2} \right\rceil$ edges

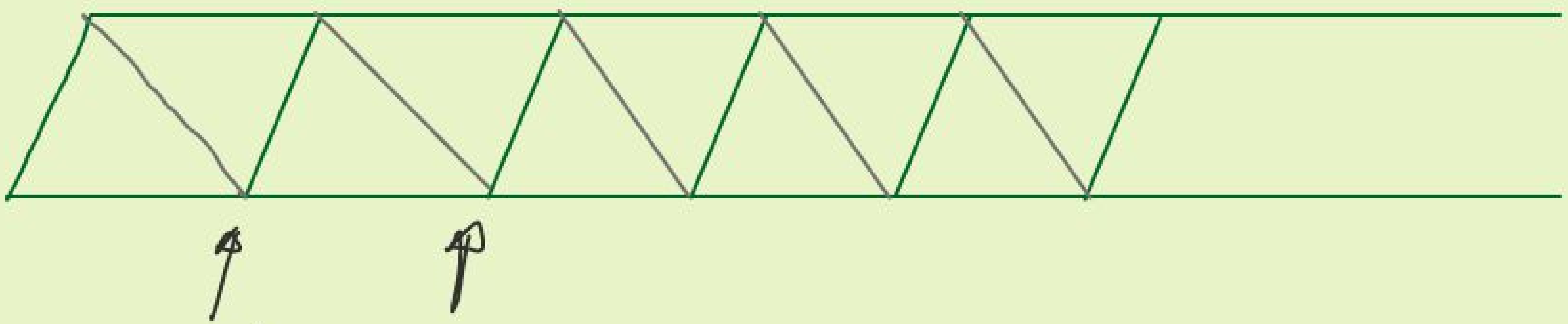
Simple graph : A graph is said to be simple if

- $\leq 1$ edge between any pair of vertices
- no self-loops

Subgraph : $H = (V', E')$ is called a subgraph of $G = (V, E)$ if $E' \subseteq E$ and $V' = \{v \mid \exists u \in V : (u, v) \in E'\}$

Bipartite subgraph : A subgraph which is bipartite.

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.



Between 2 consecutive layers 4 new edges are added. By removing 1 out of them the graph can be made bipartite.

Any simple graph has a bipartite subgraph with at least $\left\lceil \frac{|E|}{2} \right\rceil$ edges.

proof: Arbitrarily divide the vertices of the graph $G=(V,E)$ into two sets, say $X, Y$

$\forall u \in V, \quad part(u) := \begin{cases} X & \text{if } u \in X \\ Y & \text{otherwise.} \end{cases}$

$\forall u \in V \quad nbr(u) = \{ v \mid (u,v) \in E \}.$

Any simple graph has a bipartite subgraph with at least $\left\lceil \frac{|E|}{2} \right\rceil$ edges.

proof: Arbitrarily divide the vertices of the graph $G=(V,E)$ into two sets, say $X, Y$

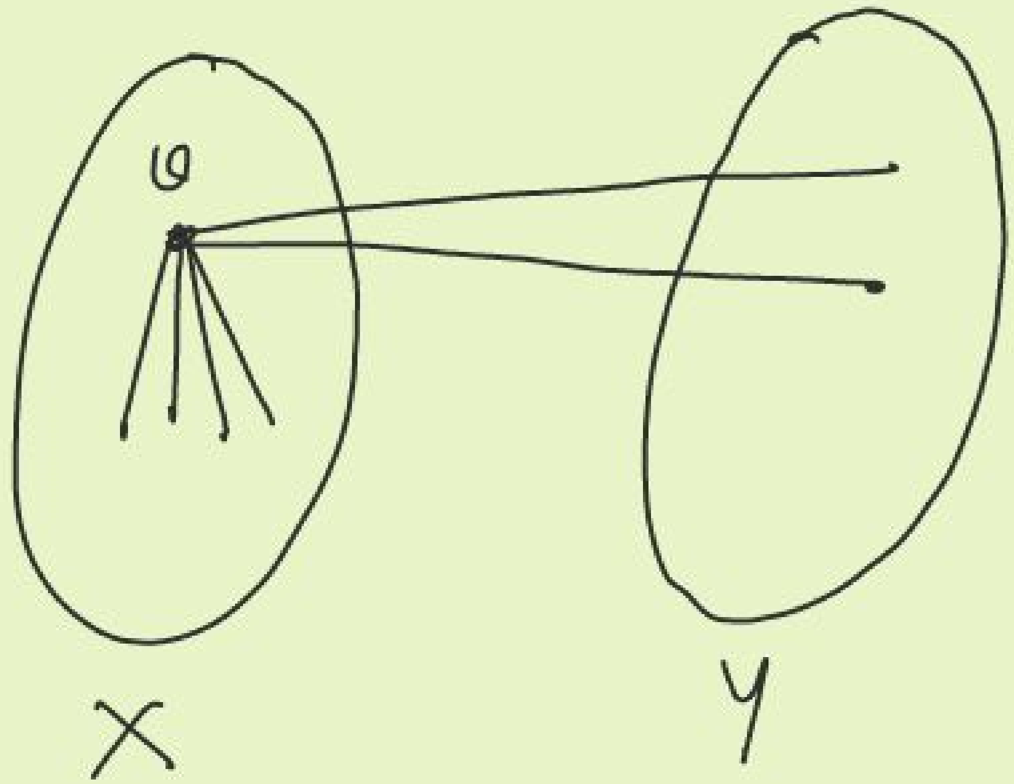
If $\exists v \in V$ s.t.

$|nbr(v) \cap part(v)| > |nbr(v) \cap V \setminus part(v)|$

then $part(v) \leftarrow V \setminus part(v)$.

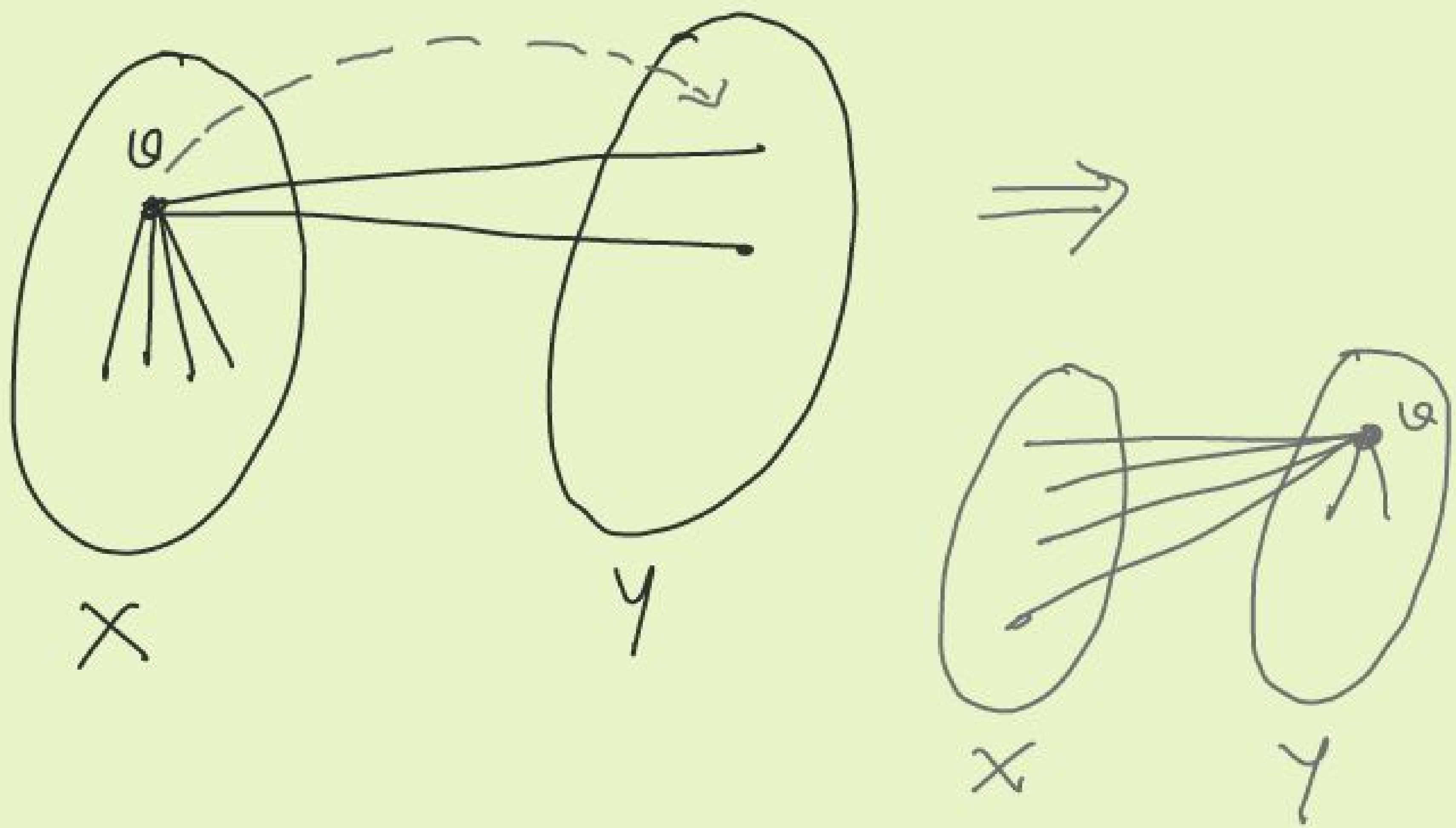
Keep doing this.

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.




$\Longrightarrow$

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.



Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.

1. The above procedure terminates:

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.

1. The above procedure terminates:

$\#$edges going across $X, Y$ always increase.

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.

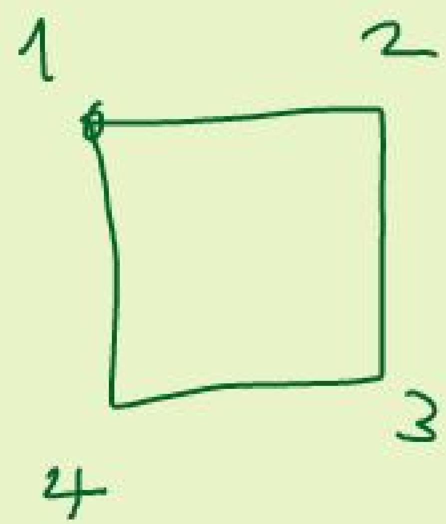
1. The above procedure terminates :
2. After the procedure terminates, the number of edges that go across $X, Y$ are at least $\left\lceil \frac{|E|}{2} \right\rceil$

Any simple graph has a bipartite subgraph with at least $\geq \left\lceil \frac{|E|}{2} \right\rceil$ edges.

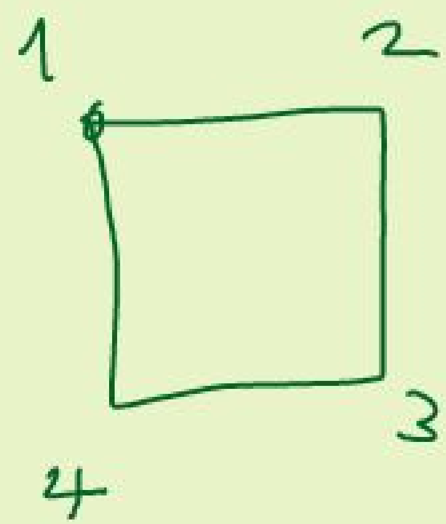
1. The above procedure terminates :
2. After the procedure terminates, the number of edges that go across $X, Y$ are at least $\left\lceil \frac{|E|}{2} \right\rceil$

   $\because$ for each vertex in part $X$ (or $Y$) at least half of its nbrs are in $V \setminus X$ (or $V \setminus Y$ respectively).

A Matching: A subset of edges in a graph is called a matching if no two edges share an endpoint.

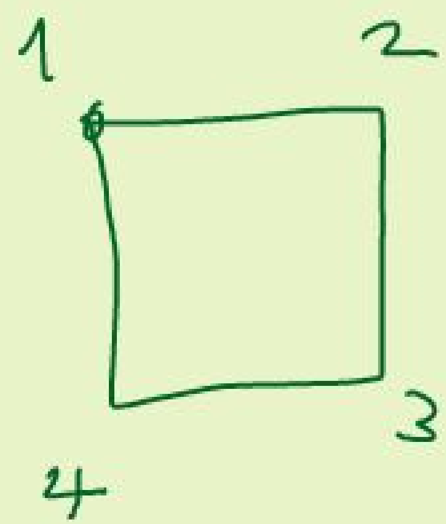


<u>A Matching</u>: A subset of edges in a graph is called a matching if no two edges share an endpoint.




$\{(1,2), (3,4)\}$

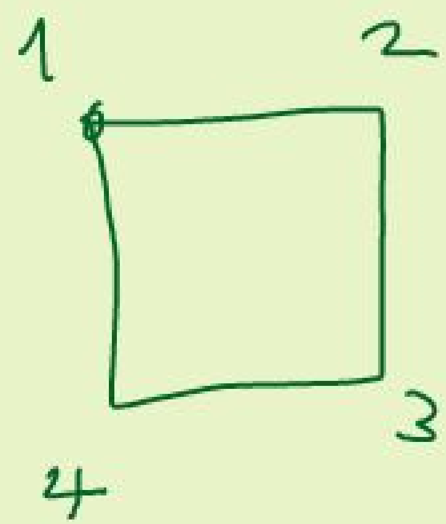
**A Matching**: A subset of edges in a graph is called a matching if no two edges share an endpoint.

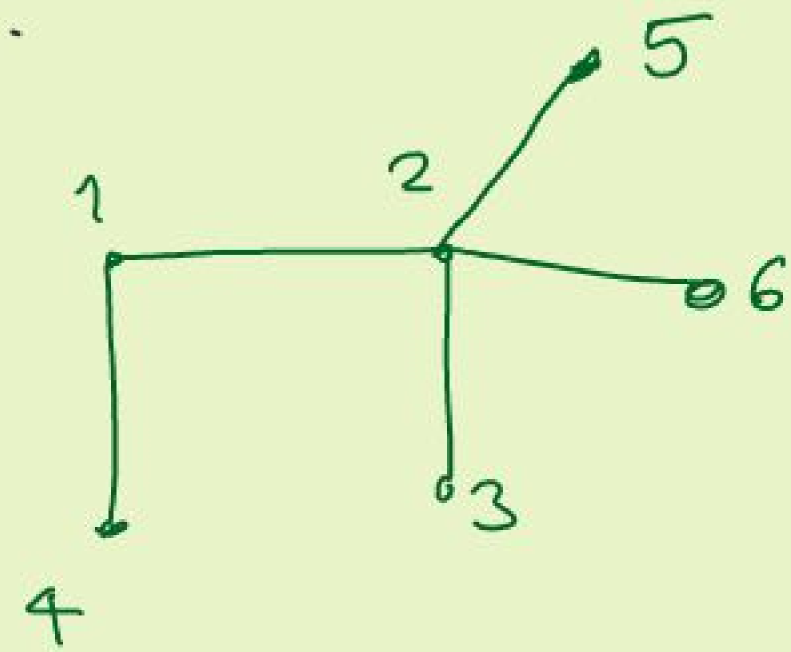



$\{(1,2), (3,4)\}$

$\{(1,4), (2,3)\}$

**A Matching**: A subset of edges in a graph is called a matching if no two edges share an endpoint.

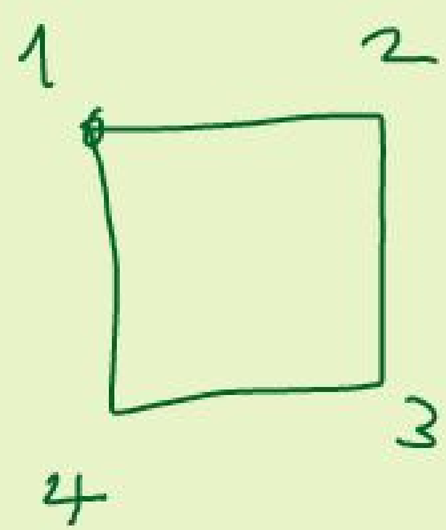



$\{(1,2), (3,4)\}$

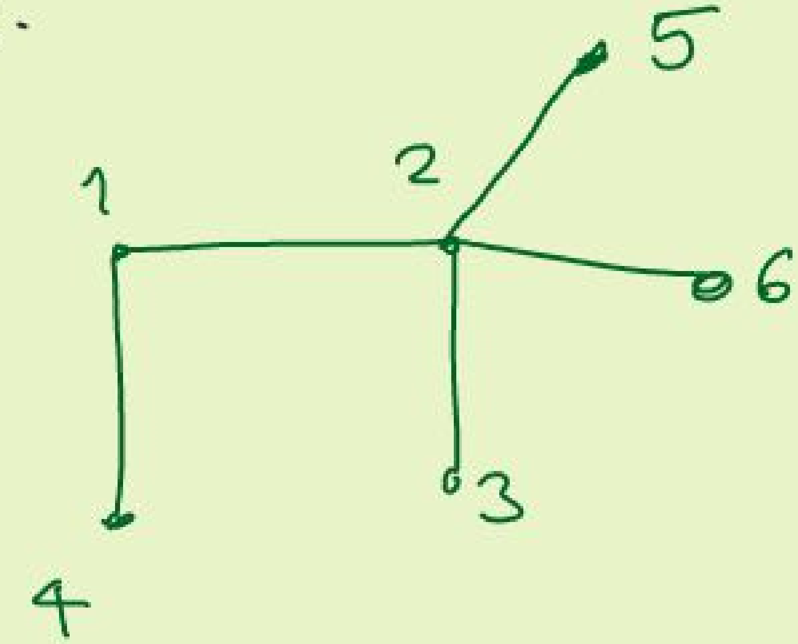
$\{(1,4), (2,3)\}$

**A Matching**: A subset of edges in a graph is called a matching if no two edges share an endpoint.




$\{(1,2), (3,4)\}$

$\{(1,4), (2,3)\}$




$\{(1,2)\}$, $\{(1,4), (2,3)\}$,

$\{(2,5)\}$, .....

**A Matching**: A subset of edges in a graph is called a matching if no two edges share an endpoint.

**A maximal matching**: A matching is called maximal if no more edges can be added to it.

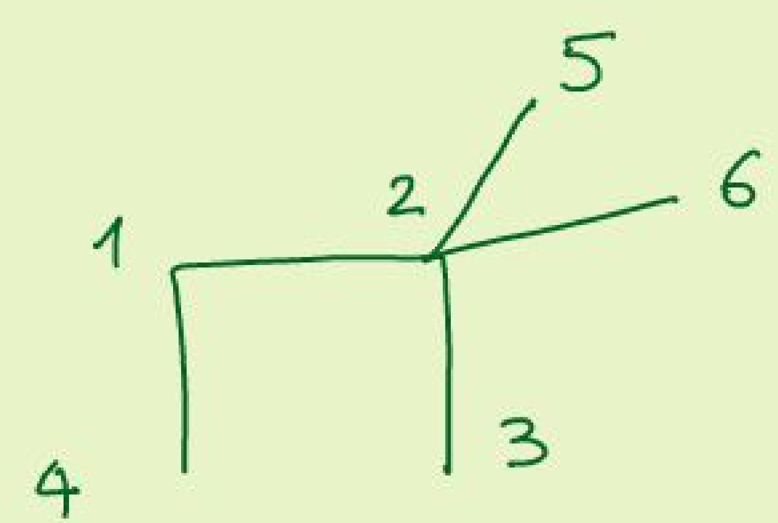
**A Matching**: A subset of edges in a graph is called a matching if no two edges share an endpoint.

**A maximal matching**: A matching is called maximal if no more edges can be added to it.

$\{(2,5)\}$,

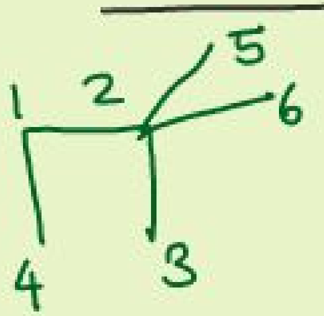
$\{(2,3), (1,4)\}$,

$\{(1,2)\}$



**Matching** : A subset of edges in a graph is called a matching if no two edges share an endpoint.

**maximal matching** : A matching is called maximal if no more edges can be added to it.

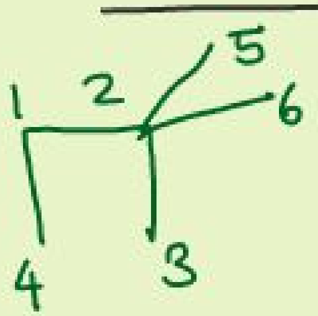



$M_1 = \{(2,5)\}$, $M_2 = \{(2,3), (1,4)\}$, $M_3 = \{(1,2)\}$

**maximum matching** : The largest cardinality maximal matching is called a maximum matching

<u>Matching</u> : A subset of edges in a graph is called a matching if no two edges share an endpoint.

<u>maximal matching</u> : A matching is called maximal if no more edges can be added to it.

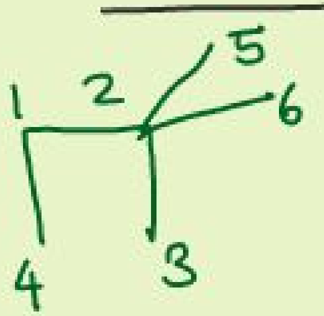



$M_1 = \{(2,5)\}$ ✗, $M_2 = \{(2,3), (1,4)\}$ ✓, $M_3 = \{(1,2)\}$ ✗

<u>maximum matching</u> : The largest cardinality maximal matching is called a maximum matching

**Matching** : A subset of edges in a graph is called a matching if no two edges share an endpoint.

**maximal matching** : A matching is called maximal if no more edges can be added to it.



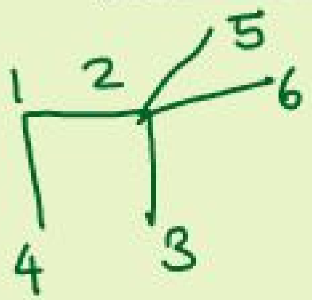

$M_1 = \{(2,5)\}$, $M_2 = \{(2,3), (1,4)\}$, $M_3 = \{(1,2)\}$

**maximum matching** : The largest cardinality maximal matching is called a maximum matching

**perfect matching** : A matching is called perfect if each vertex appears in it.

Matching : A subset of edges in a graph is called a matching if no two edges share an endpoint.

maximal matching : A matching is called maximal if no more edges can be added to it.




$M_1 = \{(2,5)\}$, $M_2 = \{(2,3), (1,4)\}$, $M_3 = \{(1,2)\}$

maximum matching : The largest cardinality maximal matching is called a maximum matching

perfect matching : A matching is called perfect if each vertex appears in it.

**Matching** : A subset of edges in a graph is called a matching if no two edges share an endpoint.

**maximal matching** : A matching is called maximal if no more edges can be added to it.

**maximum matching** : The largest cardinality maximal matching is called a maximum matching

**perfect matching** : A matching is called perfect if each vertex appears in it.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

Let $S, T$ be two sets.

$$S \Delta T = (S \cup T) \setminus (S \cap T)$$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  Suppose not. i.e. $\exists v \in M \Delta M'$ s.t $\deg(v) \geq 3$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  Suppose not. i.e. $\exists v \in M \Delta M'$ s.t $\deg(v) \geq 3$

  By PHP, at least two edges belong to the same matching $\Rightarrow \Leftarrow$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

$\therefore$ $M \Delta M'$ can consist of only cycles & paths.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

- $M \Delta M'$ has only even cycles

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.
  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.
- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.
- $M \Delta M'$ has only even cycles: for every vertex in the cycle, one of the edges incident on it is from $M$ & the other is from $M'$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

- $M \Delta M'$ has only even cycles : for every vertex in the cycle, one of the edges incident on it is from $M$ & the other is from $M'$.

  $\therefore$ only even number of edges in any cycle.

Hall's condition: Let $G = (X, Y, E)$ be a bipartite graph. There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |\,Nbr(S)\,| \geq |S|$$

Hall's condition: Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |\mathrm{Nbr}(S)| \geq |S|$$

$$\mathrm{Nbr}(S) = \{ u \mid \exists v \in S : (u,v) \in E \}$$

Set of nbrs of elements of $S$.

CS 207

Discrete Structures

Nutan Limaye

30 SEP 2013

## Last Class

1. Every simple graph has a bipartite subgraph

2. Matchings - maximal, maximum, perfect matchings.

Today:

- Perfect matchings in bipartite graphs. (Hall's condition)
- Definition of stable matchings.

**Matching** : A subset of edges in a graph is called a matching if no two edges share an endpoint.

**maximal matching** : A matching is called maximal if no more edges can be added to it.

**maximum matching** : The largest cardinality maximal matching is called a maximum matching

**perfect matching** : A matching is called perfect if each vertex appears in it.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

Let $S, T$ be two sets.

$$S \Delta T = (S \cup T) \setminus (S \cap T)$$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  Suppose not. i.e. $\exists v \in M \Delta M'$ s.t $\deg(v) \geq 3$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  Suppose not. i.e. $\exists v \in M \Delta M'$ s.t $\deg(v) \geq 3$

  By PHP, at least two edges belong to the same matching $\Rightarrow\Leftarrow$

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

$\therefore$ $M \Delta M'$ can consist of only cycles & paths.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

- $M \Delta M'$ has only even cycles

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.

  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.

- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.

- $M \Delta M'$ has only even cycles: for every vertex in the cycle, one of the edges incident on it is from $M$ & the other is from $M'$.

If $M, M'$ are two matchings in a graph then $M \Delta M'$ consist of paths or even cycles

- Every vertex in $M \Delta M'$ has degree $\leq 2$.
  $\therefore$ $M \Delta M'$ can consist of only cycles & paths.
- If $\deg(v) = 2$ in $M \Delta M'$ then exactly 1 edge is from $M$ and the other from $M'$.
- $M \Delta M'$ has only even cycles: for every vertex in the cycle, one of the edges incident on it is from $M$ & the other is from $M'$.
  $\therefore$ only even number of edges in any cycle.

**Hall's condition**: Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X : \quad |Nbr(S)| \geq |S|$$

**Hall's condition**: Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

$$Nbr(S) = \{ u \mid \exists v \in S : (u, v) \in E \}$$

Set of nbrs of elements of $S$.

<u>Hall's condition</u> : Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

- If there exists a matching $M$ in which all vertices of $X$ are matched

<u>Hall's condition</u> : Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |\,Nbr(S)\,| \geq |S|$$

- If there exists a matching $M$ in which all vertices of $X$ are matched

  Then $\forall S \subseteq X$, $M$ matches elts of $S$ to elts of $Nbr(S)$

**Hall's condition**: Let $G = (X, Y, E)$ be a bipartite graph.
There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

- If there exists a matching $M$ in which all vertices of $X$ are matched

  Then $\forall S \subseteq X$, $M$ matches elts of $S$ to elts of $Nbr(S)$

  Therefore $|Nbr(S)| \geq |S|$

**Hall's condition**: Let $G = (X, Y, E)$ be a bipartite graph.
There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X : \quad |Nbr(S)| \geq |S|$$

- If $\forall S \subseteq X : \; |Nbr(S)| \geq |S|$. then we wish to prove that no matching matches all the vertices of $X$.

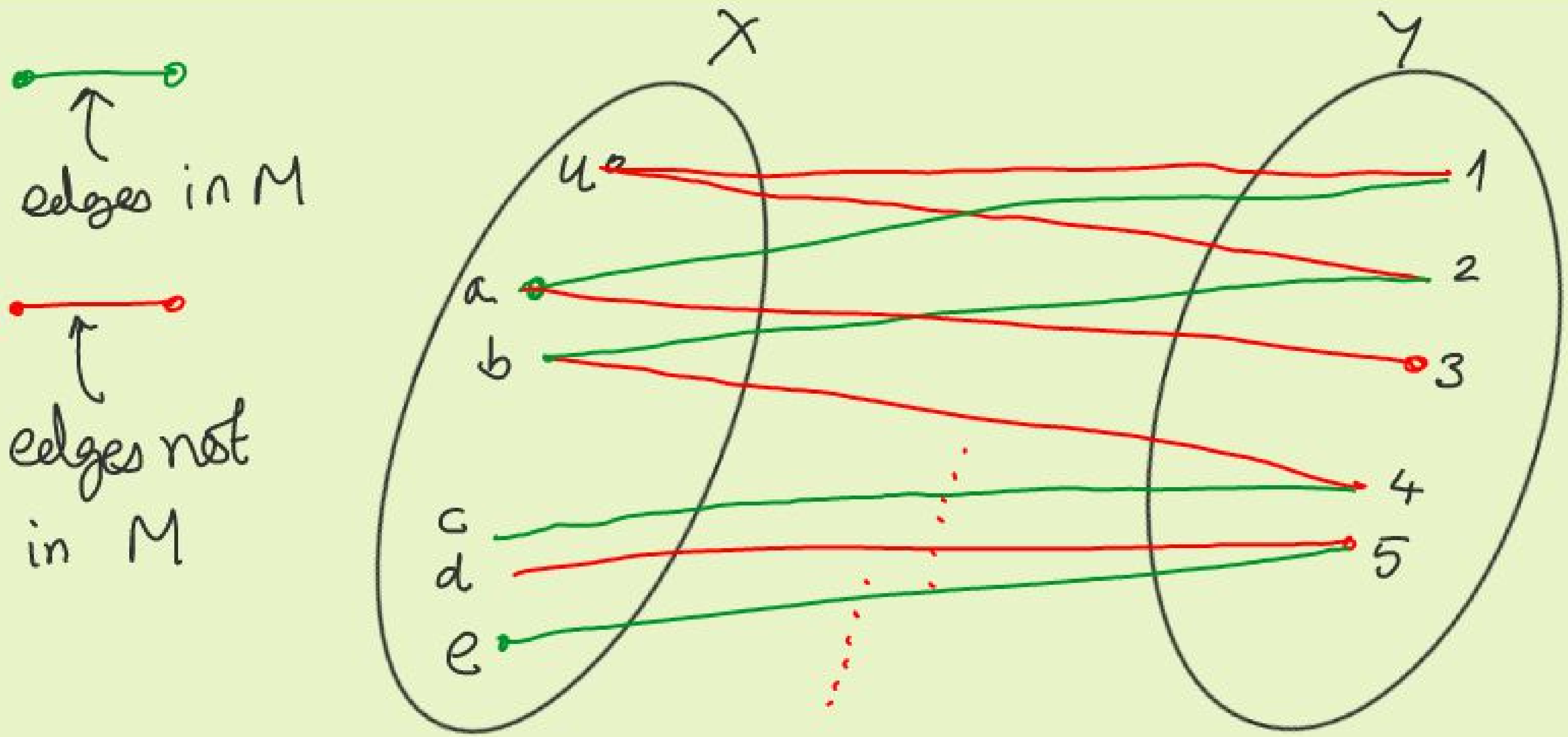
Hall's condition: Let $G = (X, Y, E)$ be a bipartite graph.

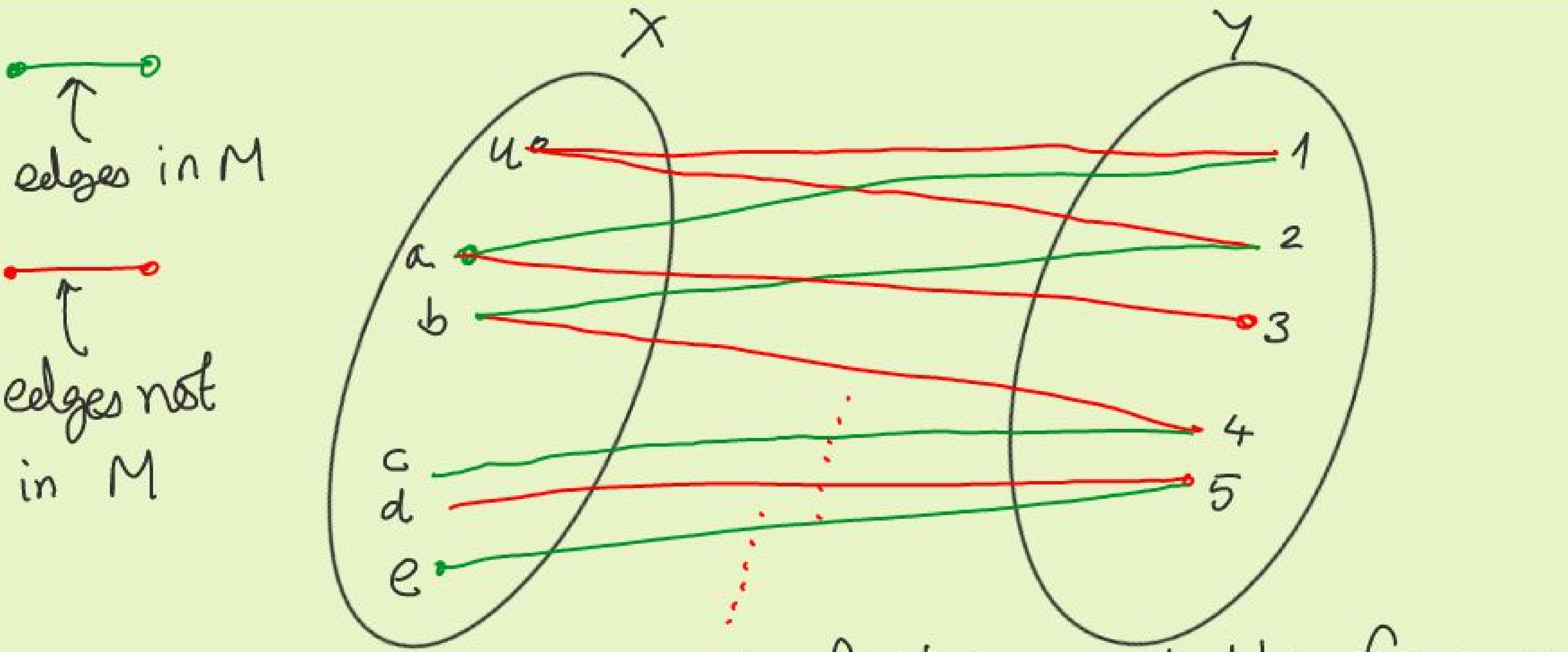
There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

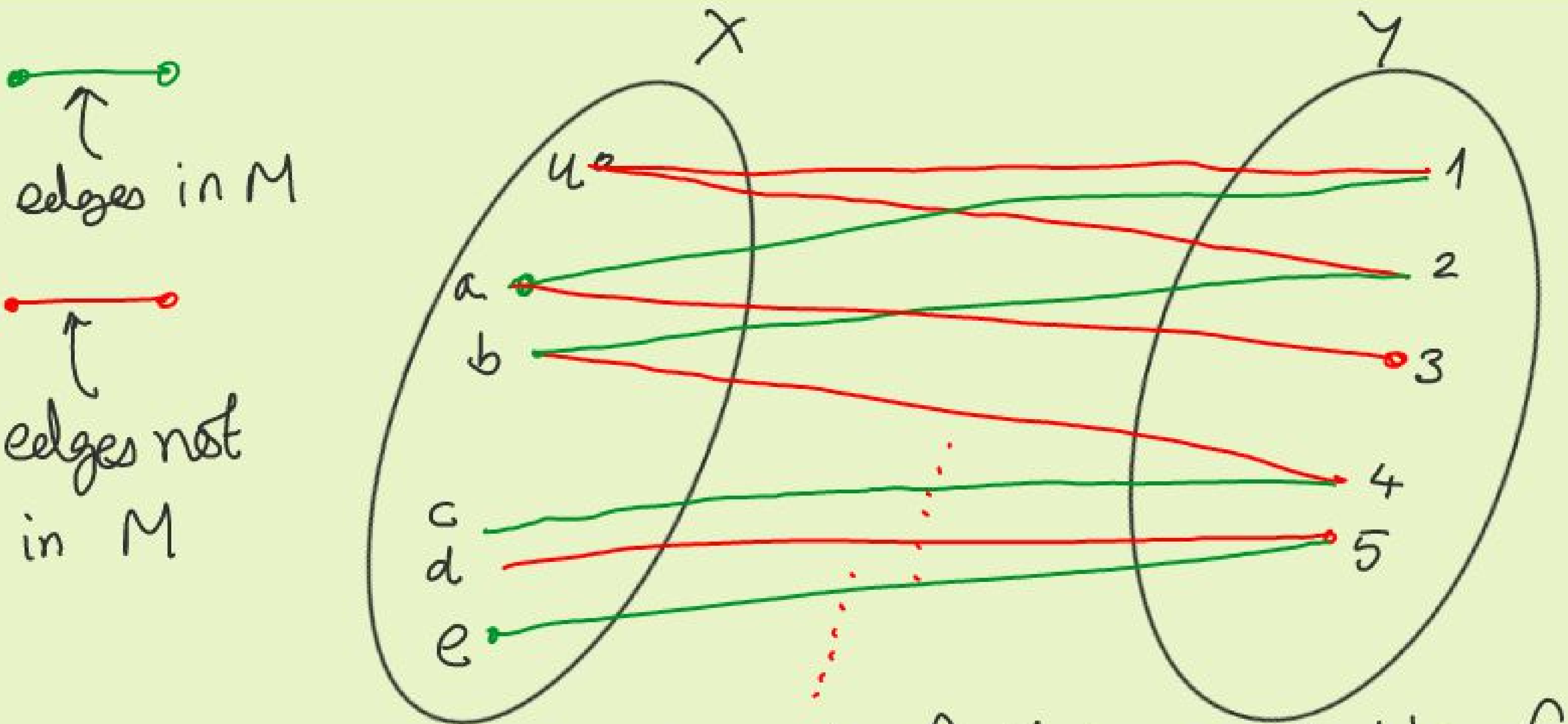
Let $M$ be a maximum matching. Suppose $M$ does not match all vertices of $X$

Let $u$ be a vertex in $X$ that is not matched by $M$.


edges in M
edges not in M
X
Y
u
a
b
c
d
e
1
2
3
4
5



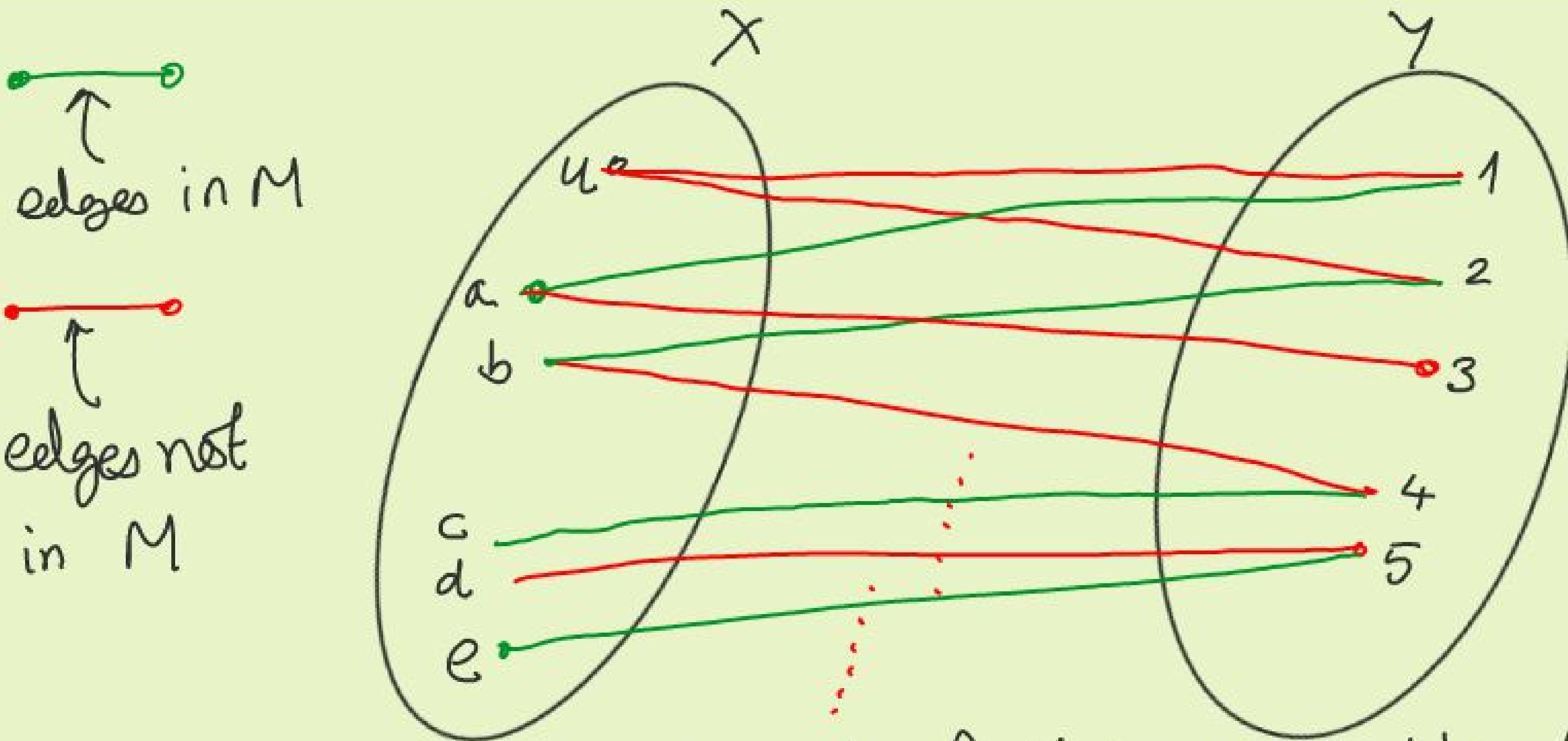

A vertex is called M-good if it is reachable from u using alternate edges from M




A vertex is called M-good if it is reachable from u using alternate edges from M

Let S be the set of M-good vertices in X

Let T be the set of M-good vertices in Y.




A vertex is called M-good if it is reachable from u using alternate edges from M

Let S be the set of M-good vertices in X

Let T be the set of M-good vertices in Y.

Observe that $|S \setminus \{u\}| = |T|$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$. $\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \le |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\because x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

Observe that $|S \setminus \{u\}| = |T|$

$\therefore$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\therefore x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

But then this path can be extended via $(x, y)$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\because x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

But then this path can be extended via $(x, y)$

$\therefore y \in T$, which is a contradiction.

## Stable Marriage Problem

Given : $n$ men and $n$ women
priority ordering of each man & woman

Output : Output a matching which has no unstable pair.

# Stable Marriage Problem

**Given** : $n$ men and $n$ women
priority ordering of each man & woman

**Output** : Output a matching which has no
unstable pair.

**Unstable pair** : Let $M$ be a matching. A pair $(x, a)$ is called unstable wrt $M$ if $x$ and $a$ prefer each other over their current pairs in $M$.

Example

| company type | | student branch | |
|---|---|---|---|
| Tech | (T) | CS | (C) |
| Finance | (F) | MECH | (M) |

Example

company type | student branch
---|---
Tech (T) | CS (C)
Finance (F) | MECH (M)

$L_T$ : $C > M$

$L_F$ : $C > M$

$L_C$ : $T > F$

$L_M$ : $T > F$

## Example

company type — Tech (T), Finance (F)

student branch — CS (C), MECH (M)

$L_T : \; C > M$

$L_F : \; C > M$

$L_C : \; T > F$

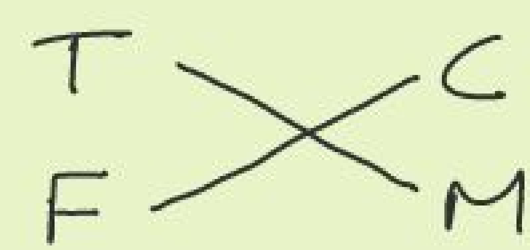
$L_M : \; T > F$

Matching 1

T —— C

F —— M

Matching 2



# CS 207

## Discrete Structures

Nutan Limaye

01 OCT 2013

# Last Class

1. Let $M, M'$ be two matchings. $M \Delta M'$ is a collection of paths or even cycles

2. Hall's condition

Today :

- Hall's condition

- Stable matchings

Hall's condition : Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geqslant |S|$$

$\underline{\text{Hall's condition}}$ : Let $G = (X, Y, E)$ be a bipartite graph.

There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

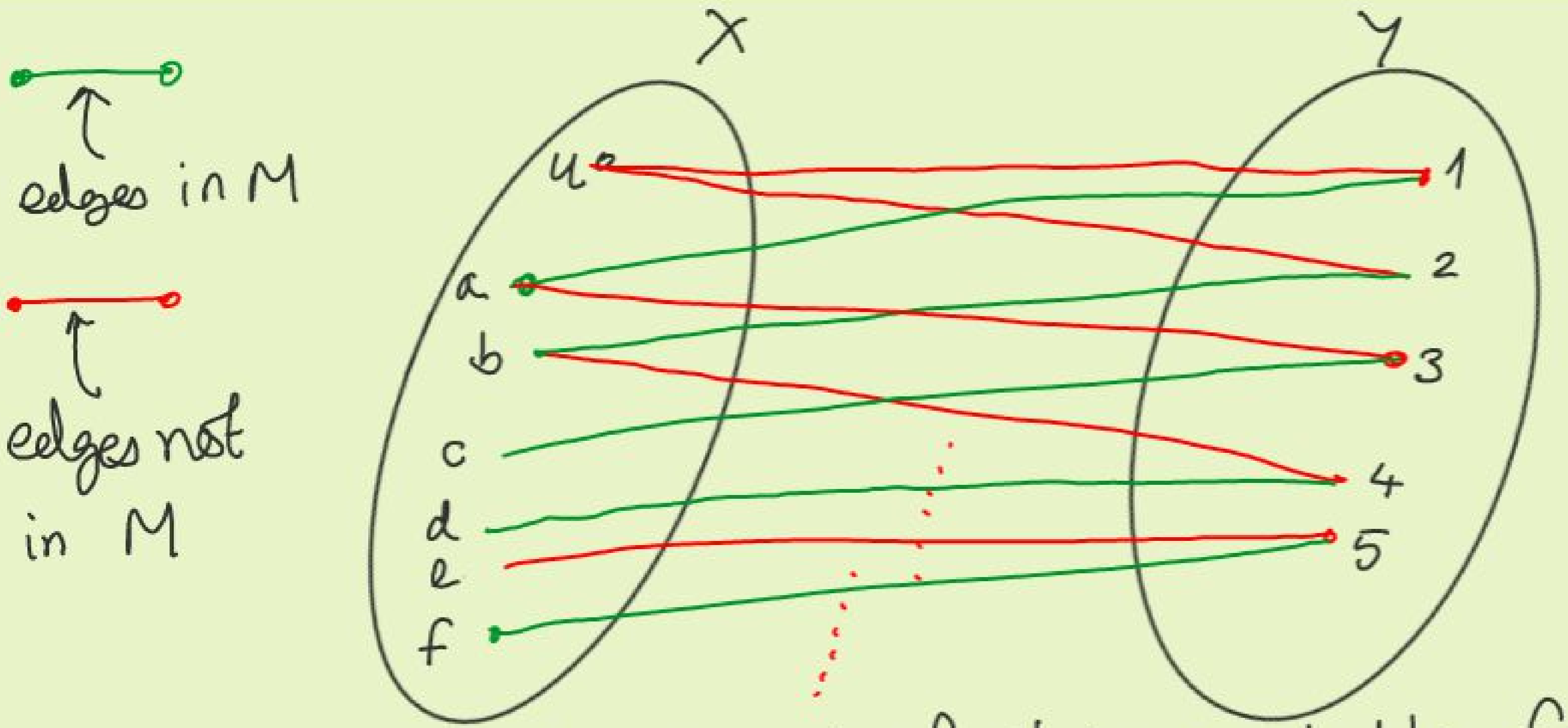
- If $\forall S \subseteq X$ : $|Nbr(S)| \geq |S|$. then we wish to prove that $\exists$ matching in which all the vertices of $X$ are matched.

**Hall's condition**: Let $G = (X, Y, E)$ be a bipartite graph. There exists a matching in which all vertices of $X$ are matched if and only if

$$\forall S \subseteq X \quad : \quad |Nbr(S)| \geq |S|$$

Let $M$ be a maximum matching. Suppose $M$ does not match all vertices of $X$

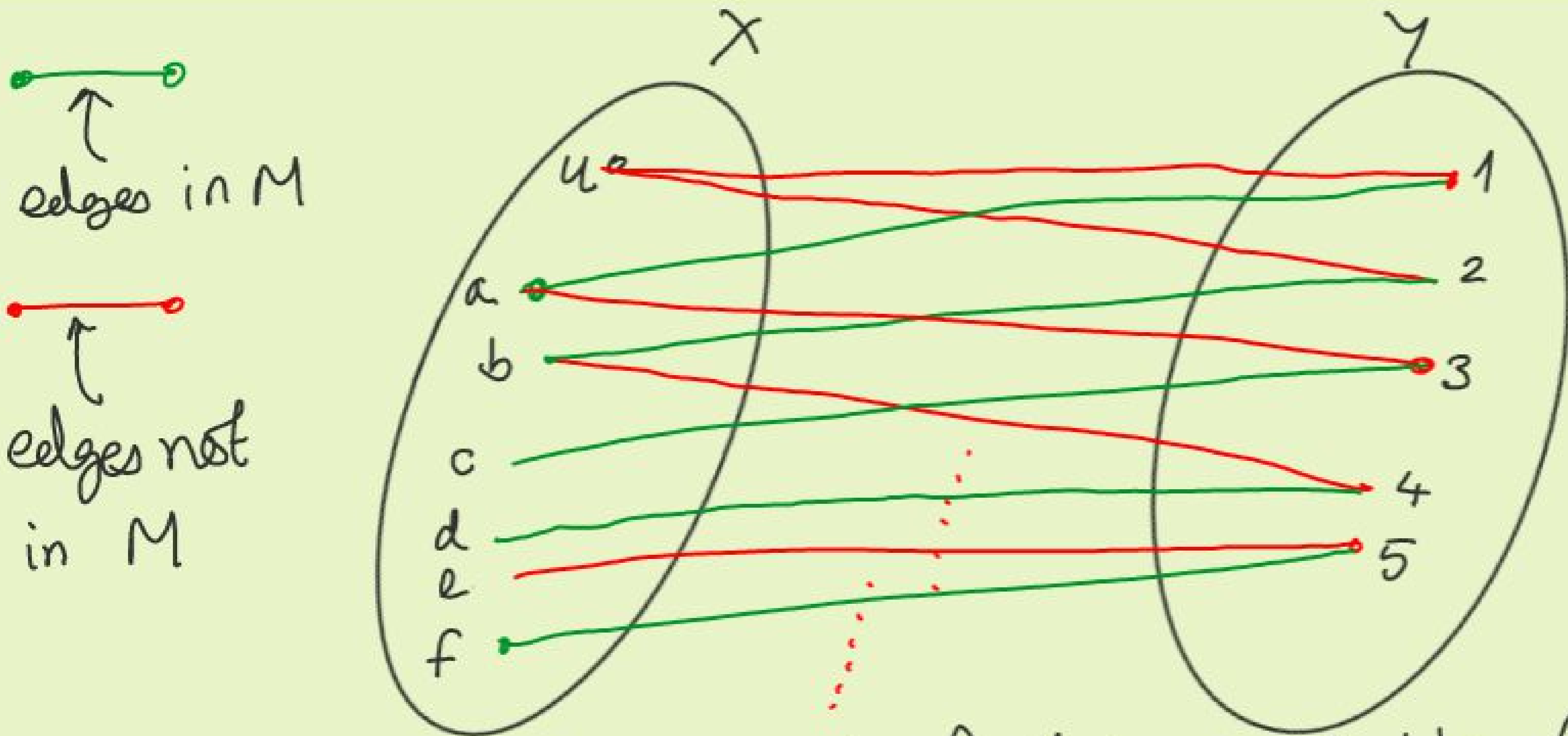
Let $u$ be a vertex in $X$ that is not matched by $M$.




A vertex is called M-good if it is reachable from u using alternate edges from M

Let S be the set of M-good vertices in X

Let T be the set of M-good vertices in Y.




Here,

$S = \{a, b, c, d, u\}$

$T = \{1, 2, 3, 4\}$

A vertex is called M-good if it is reachable from u using alternate edges from M

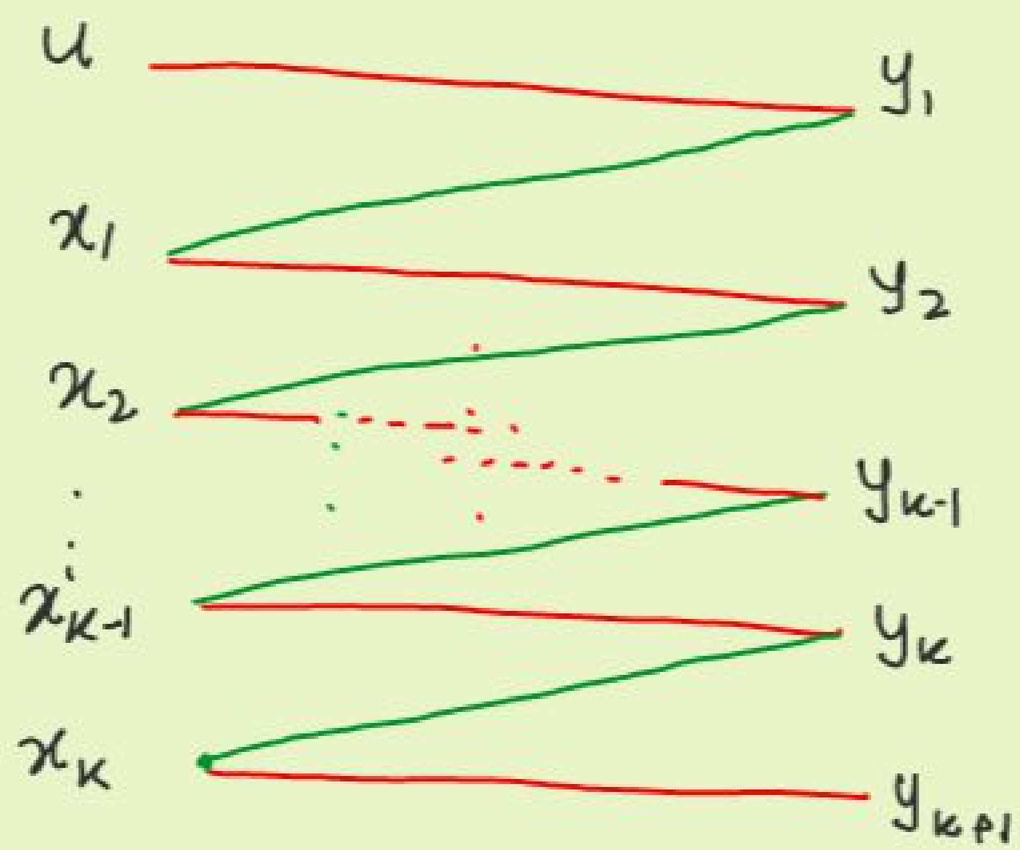
Let S be the set of M-good vertices in X

Let T be the set of M-good vertices in Y.

Observe: Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

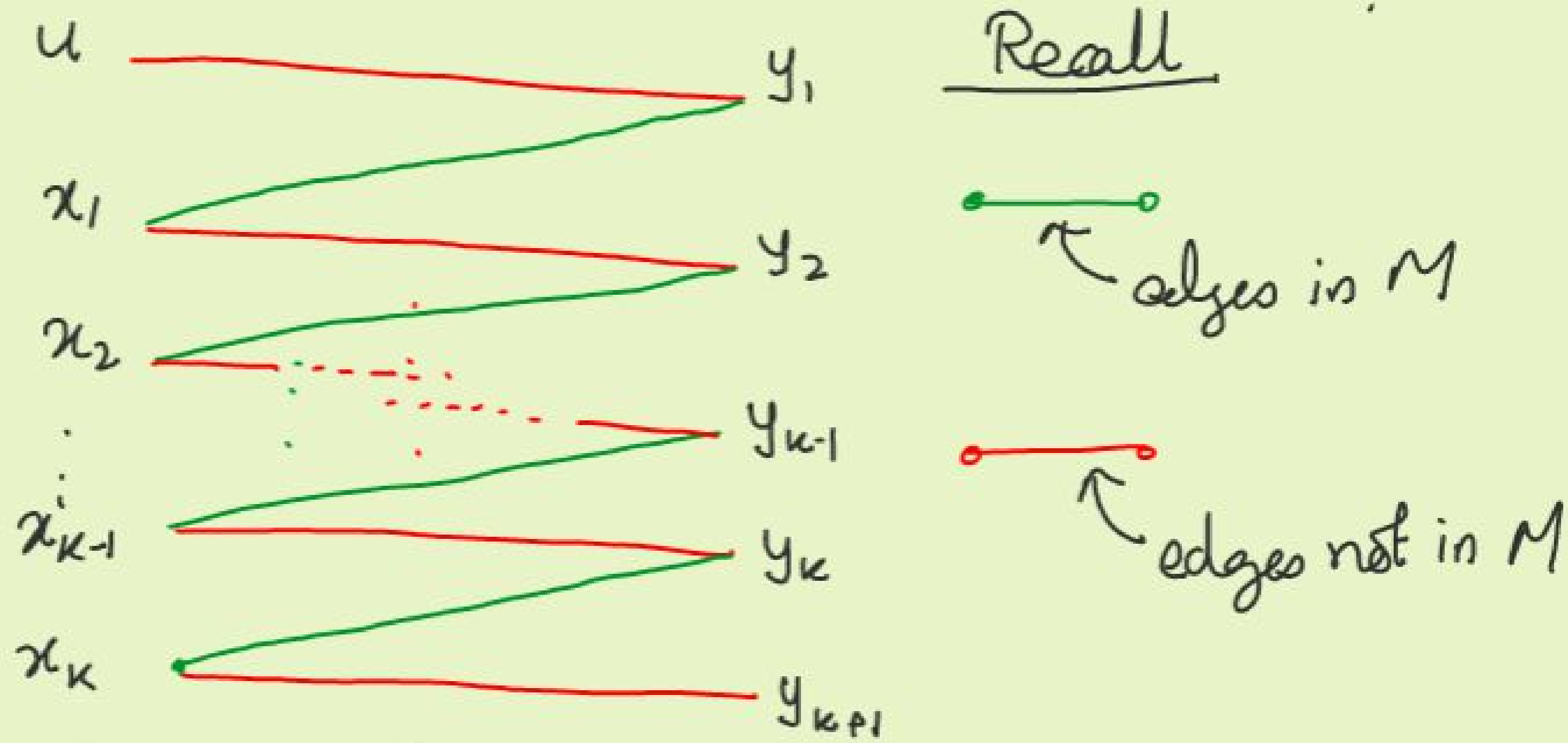
**Observe:** Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

Suppose not: That is, $\exists$ a path as follows:-



<u>Observe</u>: Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

<u>Suppose not</u>: That is, $\exists$ a path as follows:-




$$\text{Let } M' = \underbrace{\left(M \setminus \bigcup_i \{(x_i, y_i)\}\right)}_{\text{throw away green edges}} \cup \underbrace{\{(x_i, y_{i+1})\} \cup \{(u, y_1)\}}_{\text{add red edges}}$$

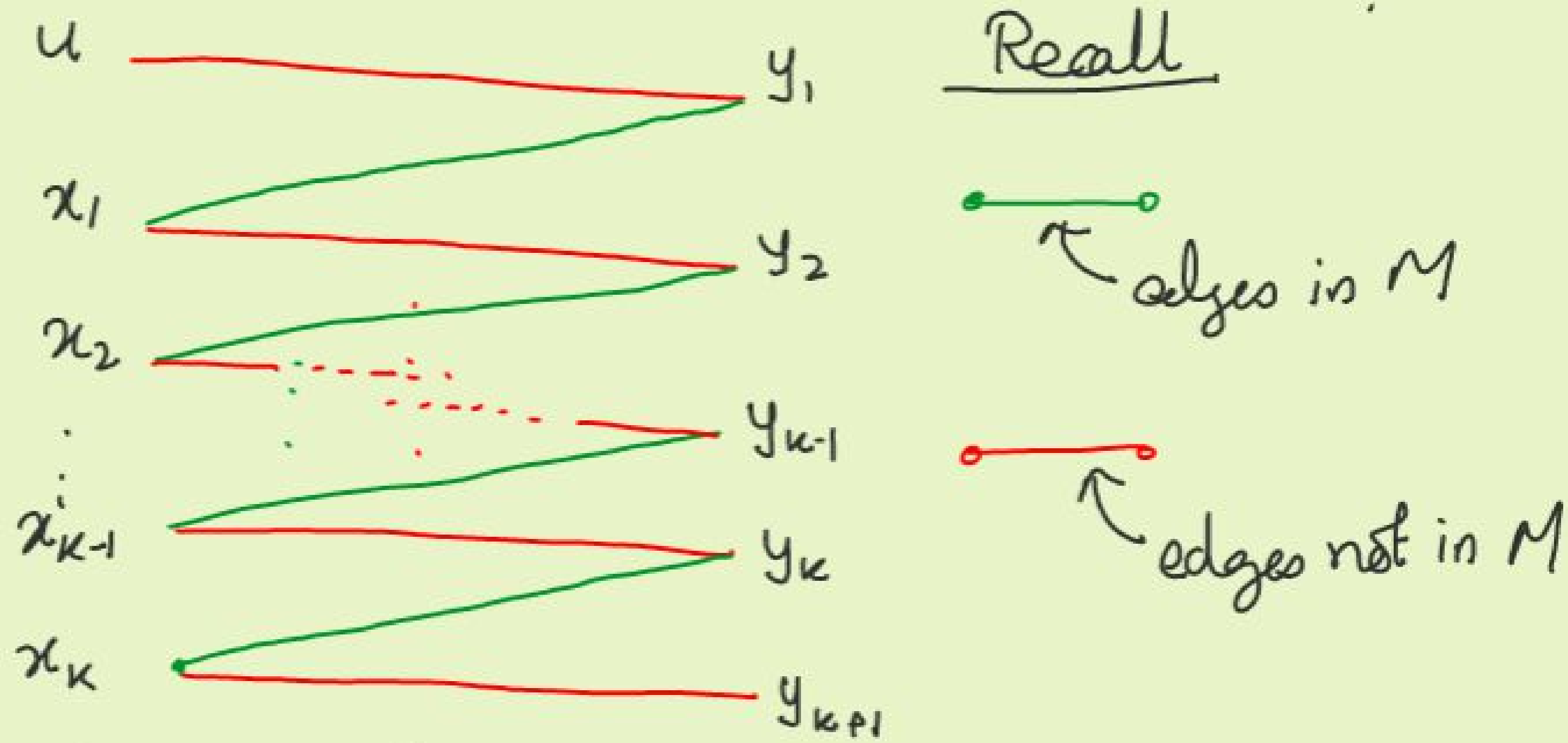
<u>Observe</u>: Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

<u>Suppose not</u>: That is, $\exists$ a path as follows:-

This contains

$k$ green edges

$k+1$ red edges

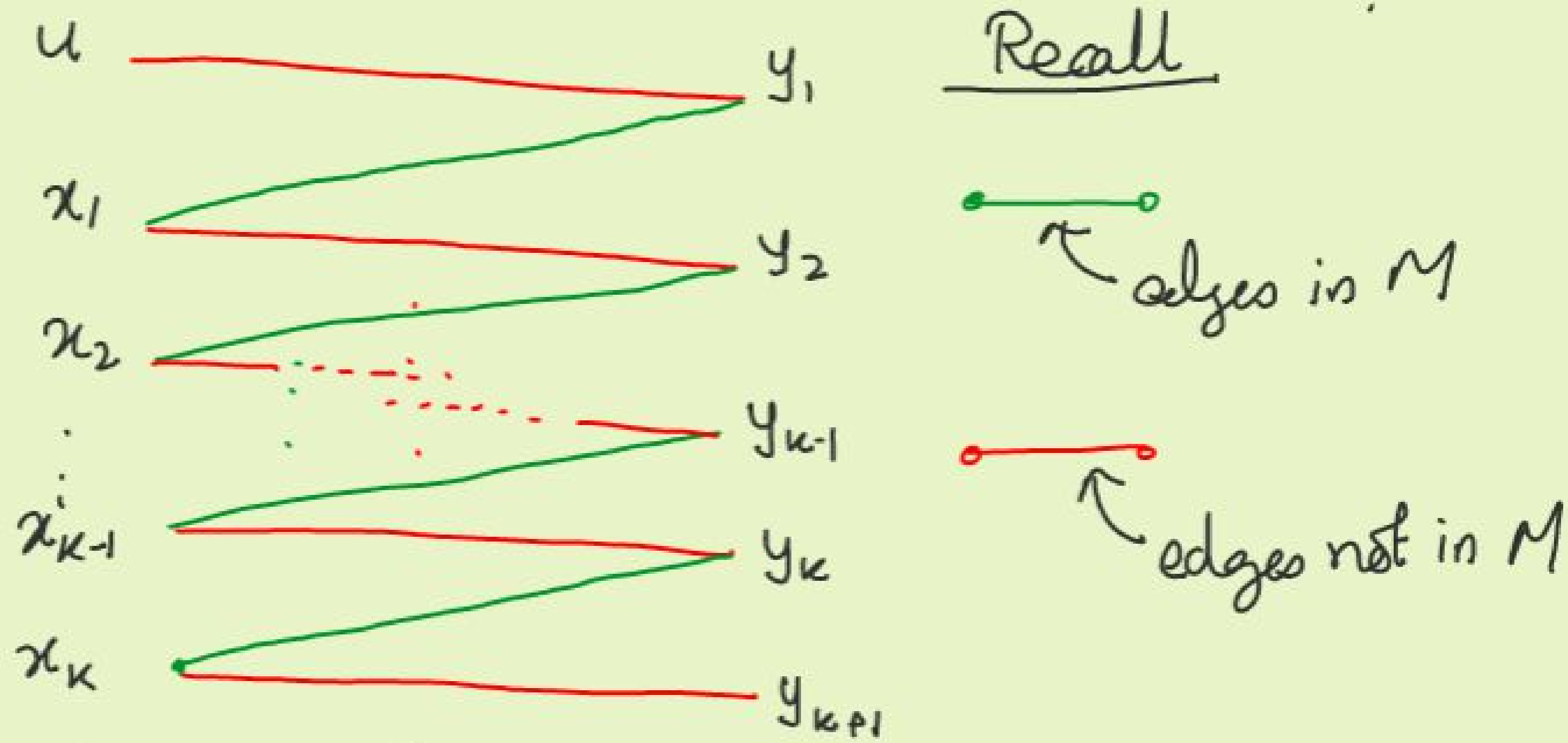



$$\text{Let } M' = \underbrace{\left(M \setminus \bigcup_i \{(x_i, y_i)\}\right)}_{\text{throw away green edges}} \underbrace{\cup \{(x_i, y_{i+1})\} \cup \{(u, y_1)\}}_{\text{add red edges}}$$

<u>Observe</u>: Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

<u>Suppose not</u>: That is, $\exists$ a path as follows:-

This contains $k$ green edges, $k+1$ red edges

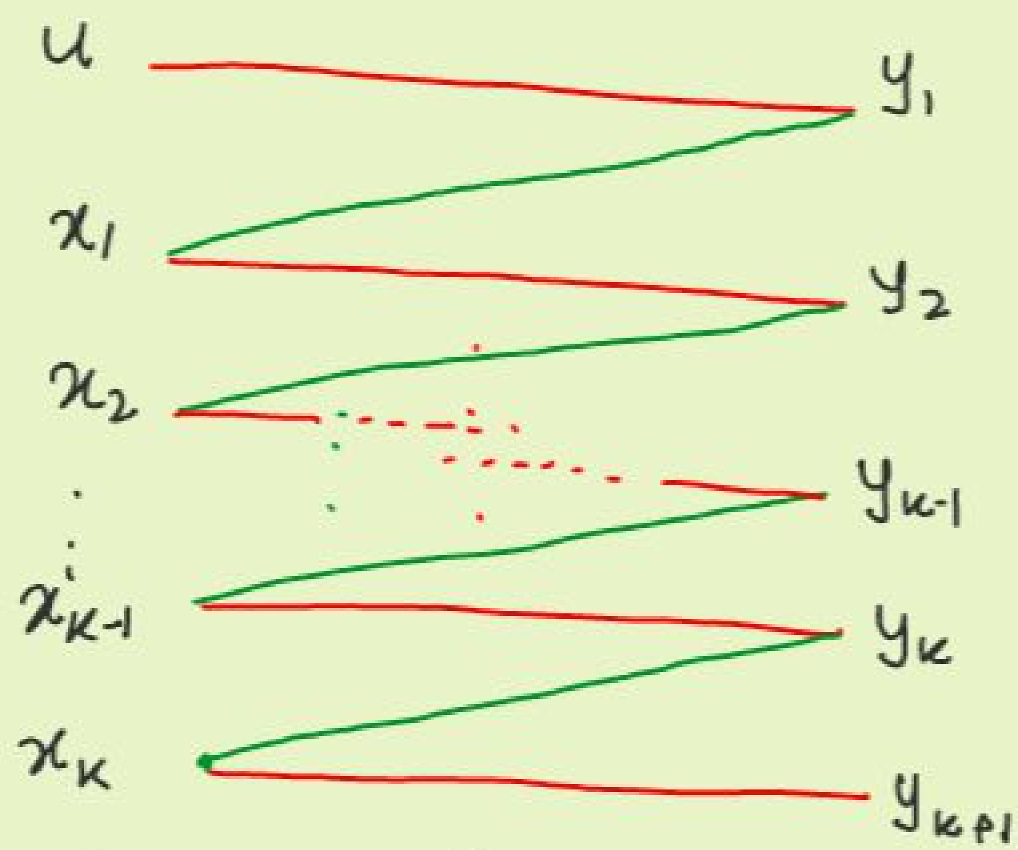



$$\text{Let } M' = \underbrace{\left(M \setminus \bigcup_i \{(x_i, y_i)\}\right)}_{\text{throw away green edges}} \cup \underbrace{\{(x_i, y_{i+1})\} \cup \{(u, y_1)\}}_{\text{add red edges}}$$

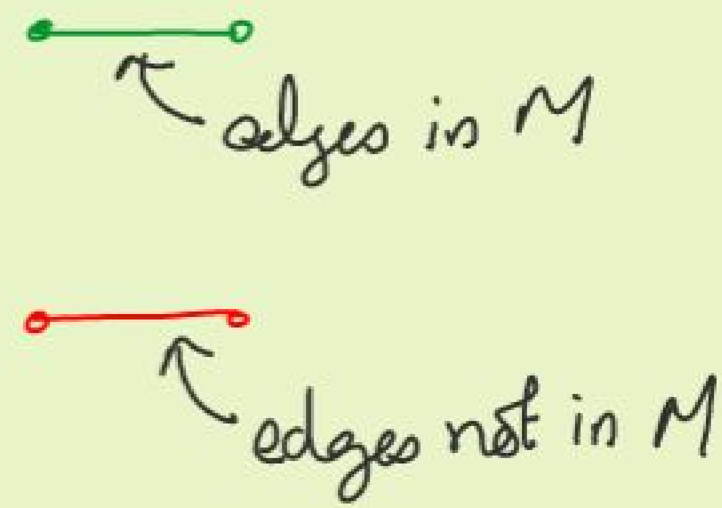
<u>Observe</u>: Any path starting from $u$ & using alternating edges from $M$ must end in a vertex in $X$.

<u>Suppose not</u>: That is, $\exists$ a path as follows:-

This contains $k$ green edges, $k+1$ red edges




<u>Recall</u>




$$\text{Let } M' = \underbrace{\Big(M \setminus \bigcup_i \{(x_i, y_i)\}\Big)}_{\text{throw away green edges}} \cup \underbrace{\{(x_i, y_{i+1})\} \cup \{(u, y_1)\}}_{\text{add red edges}}$$

Observe that $|S \setminus \{u\}| = |T|$

Observe that $|S \setminus \{u\}| = |T|$

As seen before, any path starting from $u$ which uses alt. edges from $M$ must end in $X$.

Observe that $|S \setminus \{u\}| = |T|$

As seen before, any path starting from $u$ which uses alt. edges from $M$ must end in $X$.

$\therefore$ For each vertex in $T$, $\exists$ an edge in $M$ which matches it to a vertex in $S \setminus \{u\}$

Observe that $|S \setminus \{u\}| = |T|$

$\therefore$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

As seen before, any path starting from $u$ which uses alt. edges from $M$ must end in $X$.

$\therefore$ For each vertex in $T$, $\exists$ an edge in $M$ which matches it to a vertex in $S \setminus \{u\}$

Observe that $|S \setminus \{u\}| = |T|$

$\therefore$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

(*)

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$(*)$

Observe also that $|Nbr(S)| = |T|$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

Observe that $|S\setminus\{u\}| = |T|$

$\therefore$ each vertex in $T$ is matched with a vertex in $S\setminus\{u\}$. ⊛

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to ⊛ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\therefore x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

Observe that $|S \setminus \{u\}| = |T|$

$\therefore$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

$\circledast$

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to $\circledast$ above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\therefore x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

But then this path can be extended via $(x, y)$

Observe that $|S \setminus \{u\}| = |T|$

$\because$ each vertex in $T$ is matched with a vertex in $S \setminus \{u\}$.

(*)

Observe also that $|Nbr(S)| = |T|$

$|T| \leq |Nbr(S)|$ due to (*) above.

Now suppose $\exists y \in Y$ s.t. $y \in Nbr(S)$ but $y \notin T$

Let $x \in S$ s.t. $(x, y)$ is an edge

$\because x \in S$, it is M-good.

i.e. $\exists$ a path from $u$ to $x$ using alt M edges.

But then this path can be extended via $(x, y)$

$\therefore y \in T$, which is a contradiction.

# Stable Marriage Problem

Given : $n$ men and $n$ women
priority ordering of each man & woman

Output : Output a matching which has no unstable pair.

# Stable Marriage Problem

**Given** : $n$ men and $n$ women
priority ordering of each man & woman

**Output** : Output a matching which has no unstable pair.

**Unstable pair** : Let $M$ be a matching. A pair $(x, a)$ is called unstable wrt $M$ if $x$ and $a$ prefer each other over their current pairs in $M$.

## Example

| company type | student branch |
| --- | --- |
| Tech (T) | CS (C) |
| Finance (F) | MECH (M) |

## Example

| company type | student branch |
| --- | --- |
| Tech (T) | CS (C) |
| Finance (F) | MECH (M) |

$L_T$ : $C > M$

$L_F$ : $C > M$

$L_C$ : $T > F$

$L_M$ : $T > F$

## Example

company type
Tech (T)
Finance (F)

student branch
CS (C)
MECH (M)

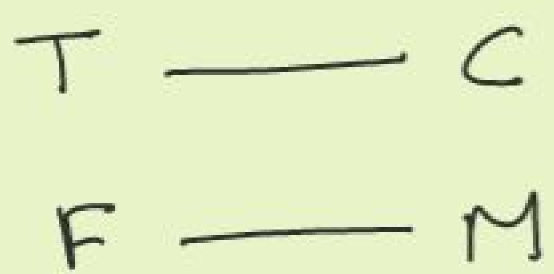
$L_T$ : $C > M$

$L_F$ : $C > M$

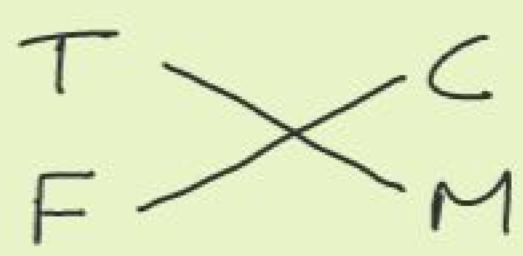
$L_C$ : $T > F$

$L_M$ : $T > F$

Matching 1




Matching 2



# CS 207

## Discrete Structures

Nutan Limaye

03 OCT 2013

# Last Class

1. Bipartite graphs have perfect matchings iff Hall's condition is satisfied

2. Stable marriage problem — Gale-Shapley algorithm.

Today :

- Analysis of GS algo.
  - Running time
  - Always outputs a perfect matching
  - Always outputs a stable matching
  - Male dominant.

## GS algorithms for Stable Marriages

Notation :· A woman/man is said to be free if she/he is not paired.

# GS algorithms for Stable Marriages

<u>Notation</u> :

- A woman/man is said to be <u>free</u> if she/he is not paired.
- A woman/man is said to be <u>engaged</u> if she/he is paired with someone during the run of the algorithm.

# GS algorithms for Stable Marriages

**Notation** :

- A woman/man is said to be **free** if she/he is not paired.
- A woman/man is said to be **engaged** if she/he is paired with someone during the run of the algorithm.
- A man and woman pair $(m, w)$ is said to be **married** if they are paired when the algorithm terminates.

# GS algorithm for Stable Marriages

While ($\exists$ a free man $m$)

{

$\boxed{1.}$ $m$ proposes to the woman $w$ who is on top of his current list

$\boxed{2.}$ if $w$ is free

 $\boxed{2.1}$ $(m, w)$ get engaged

 else /* $(m', w)$ are already engaged */

 $\boxed{2.2}$ if $L_w(m) > L_w(m')$ then

 $(m, w)$ get engaged

$\boxed{3.}$ $m$ deletes $w$ from his list

}

## Running time of GS

## Running time of GS Algo

- GS algo terminates
- GS algo runs for at most $O(n^2)$ steps

# Running time of GS

GS algo terminates :

No man can get rejected by all women

## Running time of GS

GS algo terminates :

No man can get rejected by all women

- a woman can reject only if she is engaged.

## Running time of GS

### GS algo terminates :

No man can get rejected by all women

- a woman can reject only if she is engaged.
- if a man gets rejected by the last woman on his list then all women must already be engaged

## Running time of GS

### GS algo terminates :

No man can get rejected by all women

- a woman can reject only if she is engaged.
- if a man gets rejected by the last woman on his list then all women must already be engaged
- But #women = #men & no man is engaded to two women.

# Running time of GS

GS algorithm terminates in $O(n^2)$ steps.

# Running time of GS

GS algorithm terminates in $O(n^2)$ steps.

each man proposes to any woman $\leq 1$
(Due to Step $\boxed{3}$)

## Running time of GS

GS algorithm terminates in $O(n^2)$ steps.

each man proposes to any woman $\leq 1$
(Due to Step $\boxed{3}$)

there are $n$ men & $n$ women

$\therefore$ algorithm halts in $O(n^2)$ steps

# GS algorithm outputs a perfect matching

- each man is engaged to $\leq 1$ woman.

# GS algorithm outputs a perfect matching

- each man is engaged to $\leq 1$ woman.
- algo halts when no man is free $\therefore$ each man is engaged to $\geq 1$ woman.

## GS algo outputs a stable matching

# GS algo outputs a stable matching

Suppose output of GS matches

$(m, w)$ and $(m', w')$

However, $L_m(w') > L_m(w)$ and $L_{w'}(m) > L_{w'}(m')$

i.e. $(m, w')$ is an unstable pair.

# GS algo outputs a stable matching

Suppose output of GS matches

$$(m, w) \text{ and } (m', w')$$

However, $L_m(w') > L_m(w)$ and $L_{w'}(m) > L_{w'}(m')$

i.e. $(m, w')$ is an unstable pair.

Case 1: $m$ never proposed $w'$

## GS algo outputs a stable matching

Suppose output of GS matches

$$(m, w) \text{ and } (m', w')$$

However, $L_m(w') > L_m(w)$ and $L_{w'}(m) > L_{w'}(m')$

i.e. $(m, w')$ is an unstable pair.

Case 1: $m$ never proposed $w'$

Case 2: $m$ was rejected by $w'$ when he proposed.

CS 207

Discrete Structures

Nutan Limaye

07 OCT 2013

# Last Class

1. Analysis of GS algo
   - it terminates (in $O(n^2)$ steps)
   - always outputs a stable matching

2. Male-optimal matching.

Today:

- GS outputs male-optimal matching
- How to compute a maximum matching in a bipartite graph?

# GS algorithm for Stable Marriages

While ($\exists$ a free man $m$)

{

$\boxed{1.}$ $m$ proposes to the woman $w$ who is on top of his current list

$\boxed{2.}$ if $w$ is free

  $\boxed{2.1}$ $(m, w)$ get engaged

  else /* $(m', w)$ are already engaged */

  $\boxed{2.2}$ if $L_w(m) > L_w(m')$ then

   $(m, w)$ get engaged

$\boxed{3.}$ $m$ deletes $w$ from his list

}

GS algo always outputs the male-optimal matching.

Suppose not.

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, $m$, gets rejected by his optimal match $w$.

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, $m$, gets rejected by his optimal match $w$.

$w$ rejects $m$ because she is paired with $m^*$

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, m, gets rejected by his optimal match w.

$(L_w(m) < L_w(m^*))$

w rejects m because she is paired with $m^*$

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, m, gets rejected by his optimal match w.

$(L_w(m) < L_w(m^*))$

w rejects m because she is paired with $m^*$

$\exists$ a stable matching in which $(m, w)$ are matched. In that say $(m^*, w')$ are matched.

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, m, gets rejected by his optimal match w.

$(L_w(m) < L_w(m^*))$

w rejects m because she is paired with $m^*$

(defn of optimal match)

$\exists$ a stable matching in which (m, w) are matched. In that say $(m^*, w')$ are matched.

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, $m$, gets rejected by his optimal match $\omega$.

$(L_\omega(m) < L_\omega(m^*))$

$\omega$ rejects $m$ because she is paired with $m^*$

(defn of optimal match)

$\exists$ a stable matching in which $(m, \omega)$ are matched. In that say $(m^*, \omega')$ are matched.

But $(m^*, \omega)$ form an unstable pair.

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, m, gets rejected by his optimal match $\omega$.

$(L_\omega(m) < L_\omega(m^*))$

$\omega$ rejects m because she is paired with $m^*$

(defn of optimal match)

$\exists$ a stable matching in which $(m, \omega)$ are matched. In that say $(m^*, \omega')$ are matched.

But $(m^*, \omega)$ form an unstable pair.

why?

GS algo always outputs the male-optimal matching.

Suppose not.

Let $k^{th}$ run of the while loop be the first time a man, $m$, gets rejected by his optimal match $\omega$.

$(L_\omega(m) < L_\omega(m^*))$

$\omega$ rejects $m$ because she is paired with $m^*$

(defn of optimal match)

$\exists$ a stable matching in which $(m, \omega)$ are matched. In that say $(m^*, \omega')$ are matched.

But $(m^*, \omega)$ form an unstable pair. $(\Rightarrow\Leftarrow)$

why?

Recap of module 3 :

Recap of module 3 :

- Bipartite graphs

Recap of module 3 :

- Bipartite graphs
- Bipartite $\Longleftrightarrow$ no odd cycle

Recap of module 3 :

- Bipartite graphs
- Bipartite $\Longleftrightarrow$ no odd cycle
- Every simple graph has a bipartite subgraph of size $\left\lfloor \frac{|E|}{2} \right\rfloor$

Recap of module 3:

- Bipartite graphs
- Bipartite $\iff$ no odd cycle
- Every simple graph has a bipartite subgraph of size $\lfloor \frac{|E|}{2} \rfloor$
- Bipartite matchings - Hall's condition.

Recap of module 3:

- Bipartite graphs
- Bipartite $\iff$ no odd cycle
- Every simple graph has a bipartite subgraph of size $\lfloor \frac{E}{2} \rfloor$
- Bipartite matchings - Hall's condition.
- Special matchings in bipartite graphs - stable marriage

Recap of module 3 :

- Bipartite graphs
- Bipartite $\iff$ no odd cycle
- Every simple graph has a bipartite subgraph of size $\left\lfloor \frac{|E|}{2} \right\rfloor$
- Bipartite matchings - Hall's condition.
- Special matchings in bipartite graphs - stable marriage

Does each also hint towards an algo?

## Recap of module 3 :

- Bipartite graphs
- Bipartite $\Longleftrightarrow$ no odd cycle
- Every simple graph has a bipartite subgraph of size $\lfloor \frac{E}{2} \rfloor$
- Bipartite matchings - Hall's condition.
- Special matchings in bipartite graphs - stable marriage

Does each also hint towards an algo?

- How can we find a maximum matching in a bipartite graph?

# Finding a maximum matching

# CS 207

# Discrete Structures

Nutan Limaye

08 & 10 OCT 2013

## Two classes ago

1. Male-optimality of GS algo.

Today:

- An algorithm for computing a maximum matching in a bipartite graph
- Analysis of the algorithm.

All along we will only talk about bipartite graphs

M will be used to denote a matching.

All along we will only talk about bipartite graphs

$M$ will be used to denote a matching.

A vertex will be called free wrt $M$ if it is not matched in $M$.

All along we will only talk about bipartite graphs

M will be used to denote a matching.

A vertex will be called free wrt M if it is not matched in M.

A path will be called M-alternating if it uses alternate edges from M.

All along we will only talk about bipartite graphs

M will be used to denote a matching.

A vertex will be called <u>free</u> wrt M if it is not matched in M.

A path will be called <u>M-alternating</u> if it uses alternate edges from M.

An <u>M-alternating</u> path is called M-augmenting if its two end vertices are free.

All along we will only talk about bipartite graphs

M will be used to denote a matching.

A vertex will be called <u>free</u> wrt M if it is not matched in M.

A path will be called <u>M-alternating</u> if it uses alternate edges from M.

A maximal <u>M-alternating</u> path is called M-augmenting if its two end vertices are free.

A matching M is maximum if and only if these are no M-augmenting paths

Main procedure : Input: $G = (X, Y, E)$

Output: $M^* \leftarrow$ maximum matching

1. Let $M \leftarrow \phi$
2. Let $\rho \leftarrow$ AugPath$(G, M)$
3. If $\rho = \phi$ then $M^* \leftarrow M$; output $M^*$ & halt.
4. else /* $\rho = (e_1 e_2 \cdots e_{2k+1})$ */
   - 4.1 Let $M \leftarrow \left(M \setminus \left(\bigcup_{i=1}^{k} e_{2i}\right)\right)\left(\bigcup_{j=1}^{k} e_{2j+1}\right)$
   - 4.2 Goto Step 2

AugPath($G$, $M$) (Here $G$ is a bipartite graph & $M$ is a matching)

1. Let $\overline{U} = \{u \in X \mid \exists v \in Y \text{ s.t. } (u, v) \in M\}$
2. Let $U \leftarrow X \setminus \overline{U}$.
3. If $U = \phi$ then return $\rho \leftarrow \phi$.
4. Else
   - 4.1 mark all vertices in $U$ by color "grey".
   - 4.2 While ($\exists$ a grey colored vertex in $U$)

     {

     }

AugPath(G, M) (Here G is a bipartite graph & M is a matching)

$\boxed{1.}$ Let $\overline{U} = \{u \in X \mid \exists v \in Y \text{ s.t. } (u, v) \in M\}$

$\boxed{2.}$ Let $U \leftarrow X \setminus \overline{U}$.

$\boxed{3.}$ If $U = \phi$ then return $\rho \leftarrow \phi$.

$\boxed{4.}$ Else

$\boxed{4.1}$ mark all vertices in $U$ by color "grey".

$\boxed{4.2}$ While ($\exists$ a grey colored vertex in $U$)

{ /* let $x$ be a grey vertex */

Let $T_x \leftarrow$ all vertices reachable from $x$ by M-alternating paths.

$S_x \leftarrow$ all vertices reachable from $x$ by M-alternating paths.

}

AugPath(G, M) (Here G is a bipartite graph & M is a matching)

$\boxed{1.}$ Let $\overline{U} = \{u \in X \mid \exists v \in Y \text{ s.t. } (u,v) \in M\}$

$\boxed{2.}$ Let $U \leftarrow X \setminus \overline{U}$.

$\boxed{3.}$ If $U = \phi$ then return $\rho \leftarrow \phi$.

$\boxed{4.}$ Else

$\boxed{4.1}$ mark all vertices in $U$ by color "grey".

$\boxed{4.2}$ While ($\exists$ a grey colored vertex in $U$)

{ /* let $x$ be a grey vertex */

$\boxed{4.2.1}$ If ($\exists$ a $y \in T_x$ s.t. $y$ has no M-edge incident on it)

$\boxed{4.2.2}$ then return $\rho \leftarrow$ M-alt path between $x$ & $y$.

}

AugPath(G, M) (Here G is a bipartite graph & M is a matching)

$\boxed{1.}$ Let $\overline{U} = \{u \in X \mid \exists v \in Y \text{ s.t. } (u,v) \in M\}$

$\boxed{2.}$ Let $U \leftarrow X \setminus \overline{U}$.

$\boxed{3.}$ If $U = \phi$ then return $\rho \leftarrow \phi$.

$\boxed{4.}$ Else

$\boxed{4.1}$ mark all vertices in $U$ by color "grey".

$\boxed{4.2}$ While ($\exists$ a grey colored vertex in $U$)

{ /* let $x$ be a grey vertex */

$\boxed{4.2.1}$ If ($\exists$ a $y \in T_x$ s.t. $y$ has no M-edge incident on it)

$\boxed{4.2.2}$ then return $\rho \leftarrow$ M-alt path between $x$ & $y$.

$\boxed{4.2.3}$ else mark $x$ with color black

}

$\boxed{4.3}$ return $\rho \leftarrow \phi$

## Analysis of AugPath $(G, M)$ algo

- Let $x \in U$ be an unmarked vertex s.t. $\exists y \in T_x$ s.t $y$ has no M-edges incident on it. Then M-alt path between $x$ & $y$ is M-augmenting.

## Analysis of AugPath $(G, M)$ algo

- Let $x \in U$ be an unmarked vertex s.t. $\exists y \in T_x$ s.t $y$ has no $M$-edges incident on it. Then $M$-alt path between $x$ & $y$ is $M$-augmenting.
- $M$ is not maximum iff $\exists x \in U$, $\exists y \in T_x$ s.t $y$ has no $M$-edges incident on it.

## Analysis of AugPath $(G, M)$ algo

- Let $x \in U$ be an unmarked vertex s.t. $\exists y \in T_x$ s.t $y$ has no $M$-edges incident on it. Then $M$-alt path between $x$ & $y$ is $M$-augmenting.
- $M$ is not maximum iff $\exists x \in U$, $\exists y \in T_x$ s.t $y$ has no $M$-edges incident on it.
- If $U = \phi$ then $M$ is a maximum matching

## Analysis of AugPath(G, M) algo

- Let $x \in U$ be an unmarked vertex s.t. $\exists y \in T_x$ s.t. $y$ has no M-edges incident on it. Then M-alt path between $x$ & $y$ is M-augmenting.
- M is not maximum iff $\exists x \in U$, $\exists y \in T_x$ s.t. $y$ has no M-edges incident on it.
- If $U = \phi$ then M is a maximum matching

(It is possible that $U = \phi$ but M is still maximum.)

## Analysis of AugPath $(G, M)$ algo

- Let $x \in U$ be an unmarked vertex s.t. $\exists y \in T_x$ s.t $y$ has no $M$-edges incident on it. Then $M$-alt path between $x$ & $y$ is $M$-augmenting.
- $M$ is not maximum iff $\exists x \in U, \exists y \in T_x$ s.t $y$ has no $M$-edges incident on it.
- If $U = \phi$ then $M$ is a maximum matching
- For each vertex in $U$, every edge of the graph is visited exactly once. $\therefore$ AugPath $(G, M)$ runs in time $O(n \cdot m)$, where $m \leftarrow$ #edges in $G$.

CS 207

Discrete Structures

Nutan Limaye

14 OCT 2013

Last Class :

- Augmenting path algorithm for computing maximum matching
- Analysis of the algorithm.

Today:

- Directed graphs and tournaments
- Existence of a king
- Existence of a global winner.

## Directed graphs

A tournament graph: A directed graph $G = (V, E)$ is called a tournament graph if $\forall x, y \in V$

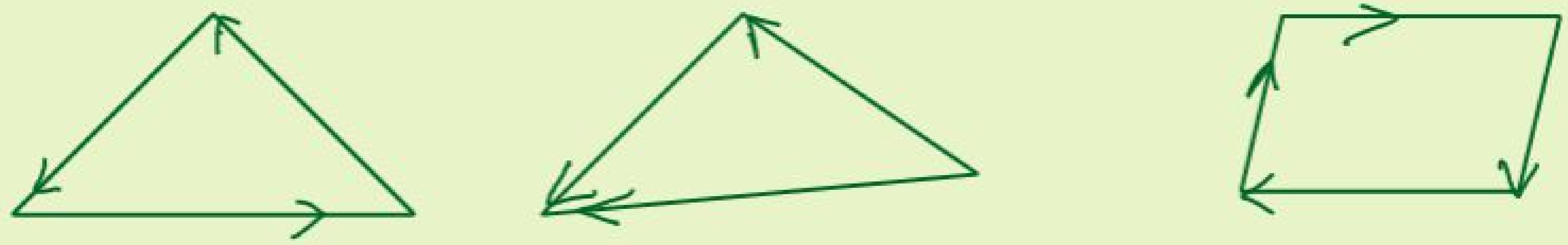
- either $(x, y)$ edge is in $E$
- or $(y, x)$ edge is in $E$
- but not both.

## Directed graphs

A tournament graph: A directed graph $G = (V, E)$ is called a tournament graph if $\forall x, y \in V$

- either $(x, y)$ edge is in $E$
- or $(y, x)$ edge is in $E$
- but not both.

Examples:

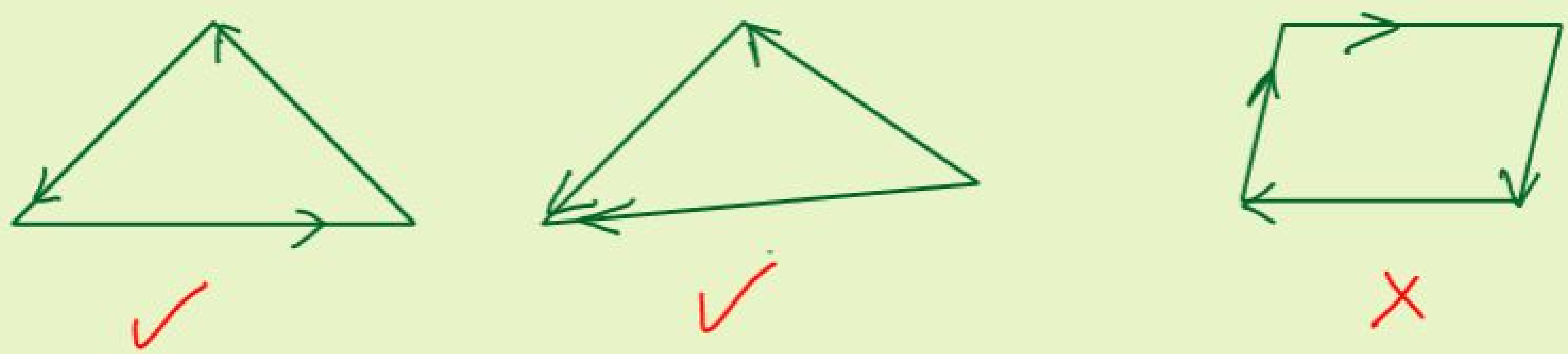


## Directed graphs

A tournament graph: A directed graph $G = (V, E)$ is called a tournament graph if $\forall x, y \in V$

- either $(x, y)$ edge is in $E$
- or $(y, x)$ edge is in $E$
- but not both.

Examples:



A vertex $x$ is said <u>to defeat</u> vertex $y$ if the edge between $x$ & $y$ is directed from $x$ to $y$.

A vertex $x$ is said to defeat vertex $y$ if the edge between $x$ & $y$ is directed from $x$ to $y$.

Let $U \subseteq V$. A vertex $x \in V$ is said to be a winner with respect to $U$ if $\forall u \in U$ $x$ defeats $u$.

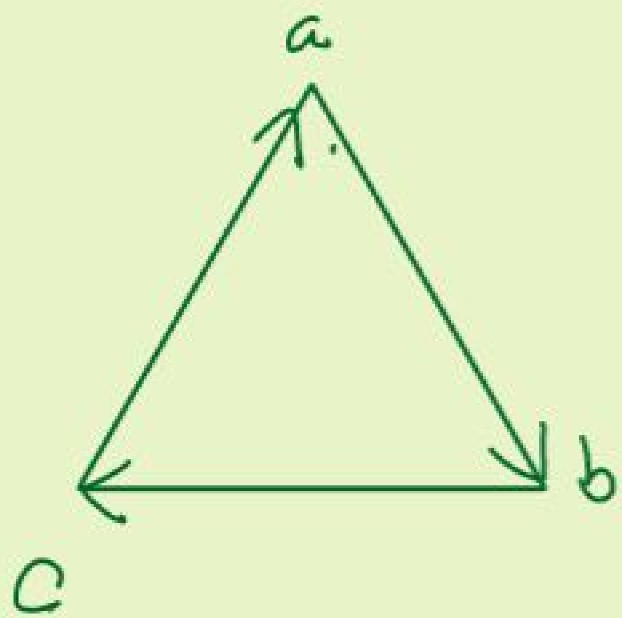
A vertex $x$ is said to defeat vertex $y$ if the edge between $x$ & $y$ is directed from $x$ to $y$.

Let $U \subseteq V$. A vertex $x \in V$ is said to be a winner with respect to $U$ if $\forall u \in U$ $x$ defeats $u$.

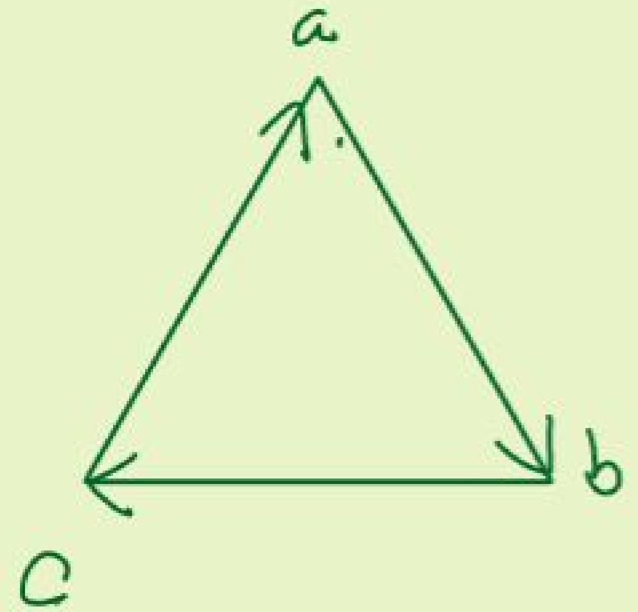
A vertex $x \in V$ is said to be a king if $\forall y \in V$

- either $x$ defeats $y$
- or $x$ defeats a vertex, say $w$, which defeats $y$.

Examples:

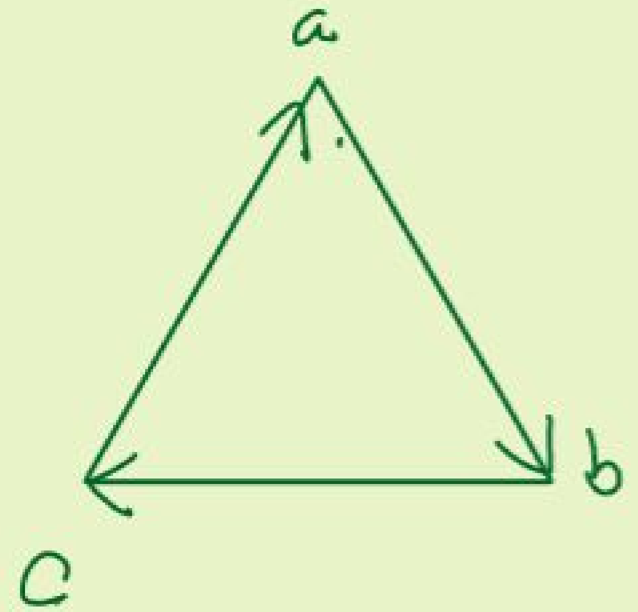


Examples:




Which vertices are kings?

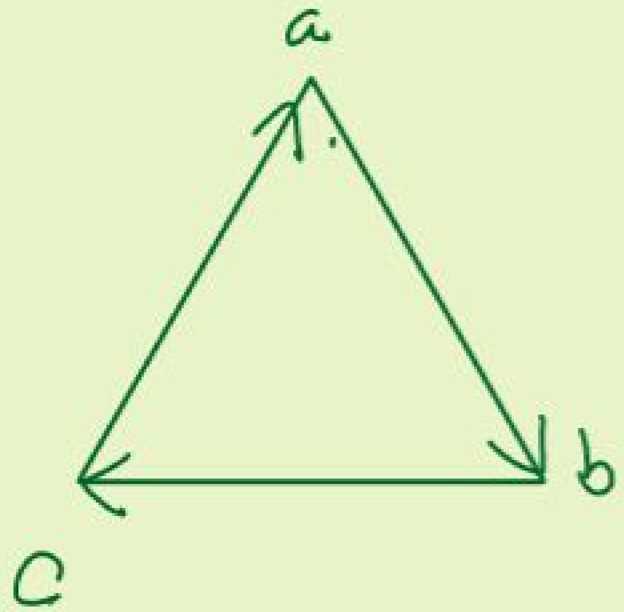
Examples:




Which vertices are kings?

— all vertices

Examples :

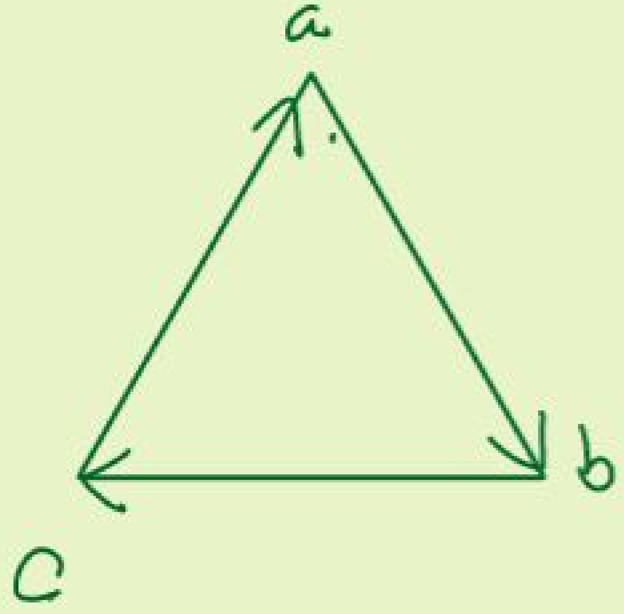



Which vertices are kings?

— all vertices

Is there a winner wrt $\{a, b\}$ ?

Examples:



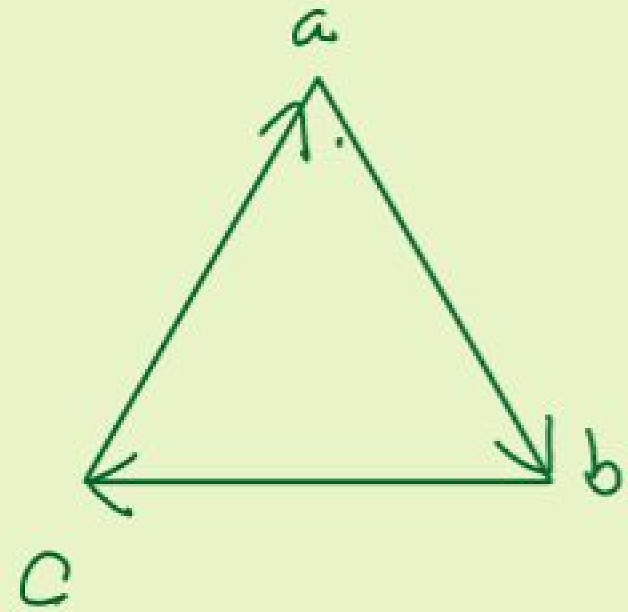

Which vertices are kings?

— all vertices

Is there a winner wrt $\{a, b\}$?

— no.

## Examples:




Which vertices are kings?

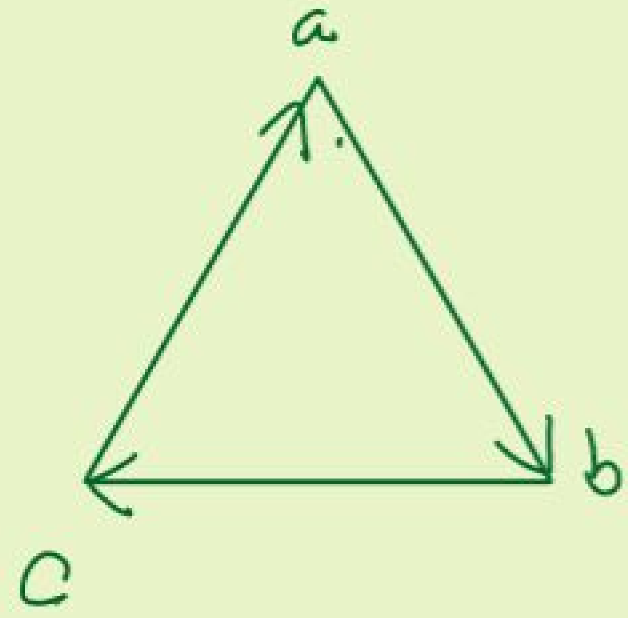
- all vertices

Is there a winner wrt $\{a, b\}$?

- no.

Is there a winner wrt $\{a\}$?

## Examples :




Which vertices are kings?

- all vertices

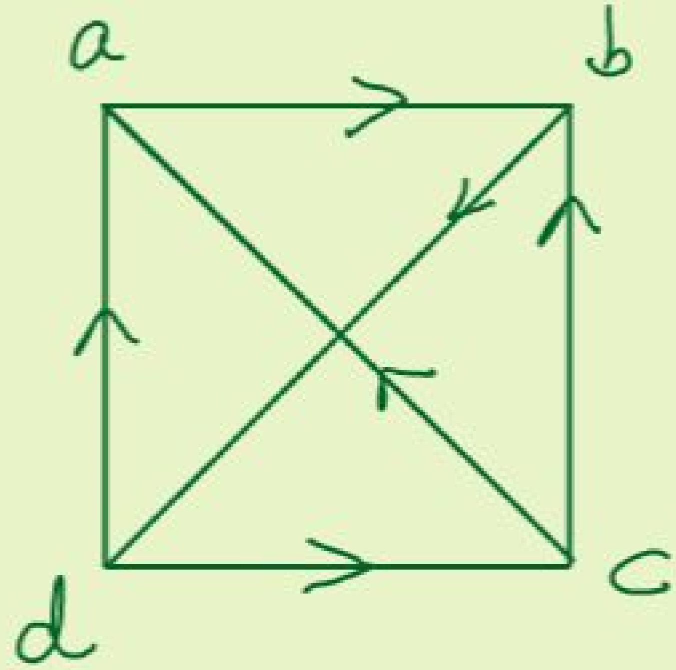
Is there a winner wrt $\{a, b\}$ ?

- no.

Is there a winner wrt $\{a\}$ ?

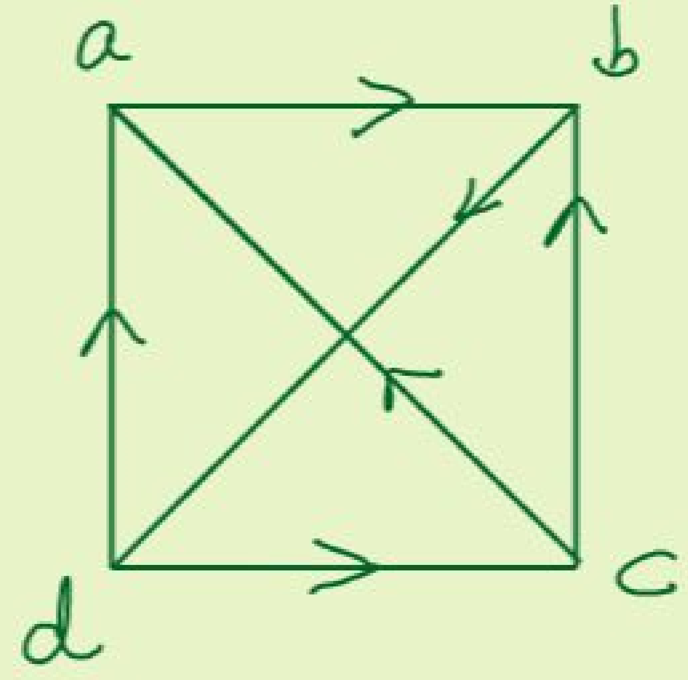
- yes.

Examples :



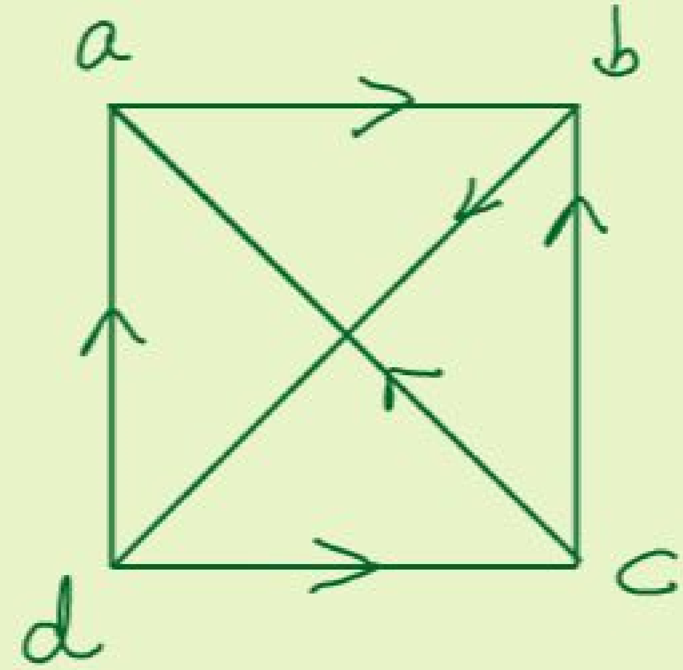
Examples:

Which vertices are kings?



Examples:

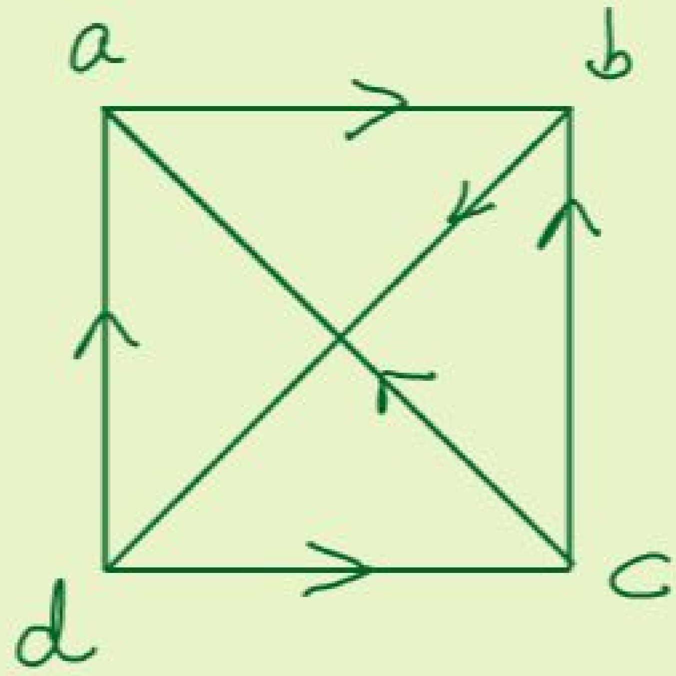
Which vertices are kings?




Is there a winner wrt $\{a, b\}$?

Examples:

Which vertices are kings?

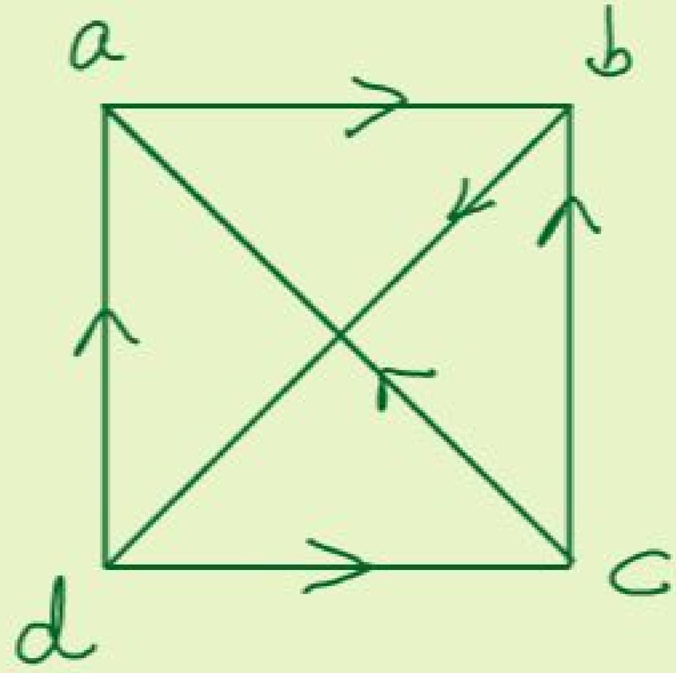



Is there a winner wrt $\{a, b\}$?

— yes.

Examples:

Which vertices are kings?



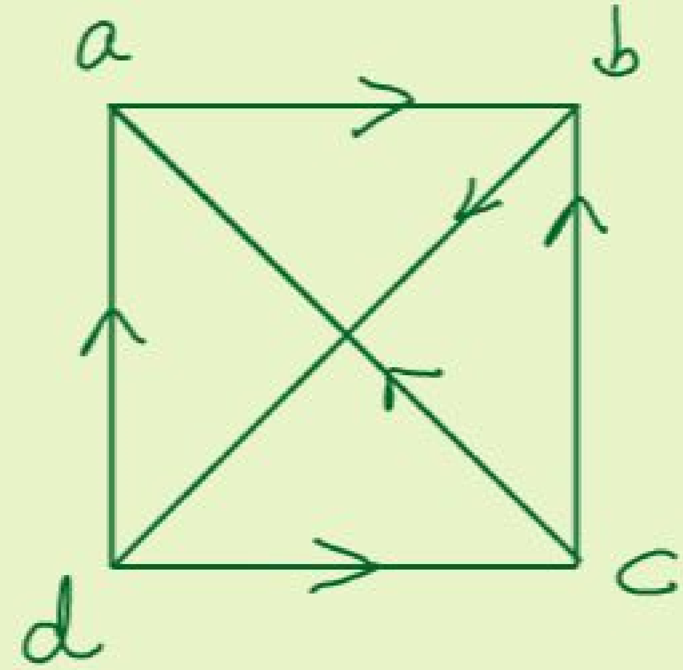

Is there a winner wrt $\{a, b\}$?

— yes.

Is there a winner wrt $\{b, c\}$

## Examples:

Which vertices are kings?




Is there a winner wrt $\{a, b\}$?

— yes.

Is there a winner wrt $\{b, c\}$

— no.

Every tournament has a king.

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{y \mid x \text{ defeats } y\}$

$N(D_x) = \{z \mid \exists y \in D_x \text{ and } y \text{ defeats } z\}$

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{ y \mid x \text{ defeats } y \}$

$N(D_x) = \{ z \mid \exists y \in D_x \text{ and } y \text{ defeats } z \}$

If $x$ itself is a king then we are done

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{ y \mid x \text{ defeats } y \}$

$N(D_x) = \{ z \mid \exists y \in D_x \text{ and } y \text{ defeats } z \}$

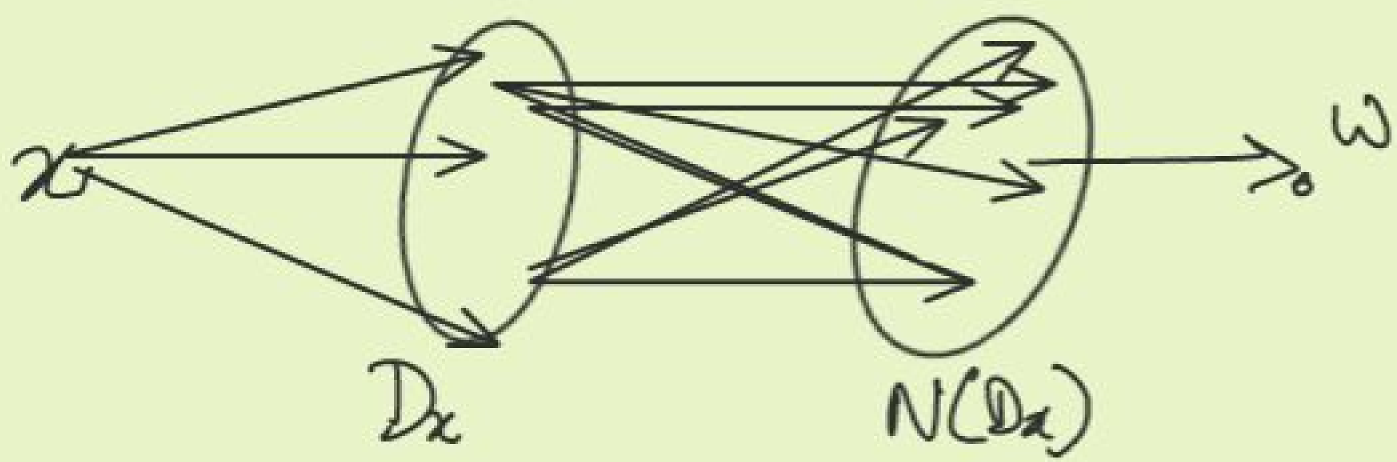
Let $w \in V$ s.t. $w \notin D_x$, $w \notin N(D_x)$

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{ y \mid x \text{ defeats } y \}$

$N(D_x) = \{ z \mid \exists y \in D_x \text{ and } y \text{ defeats } z \}$




$w \notin D_x$.

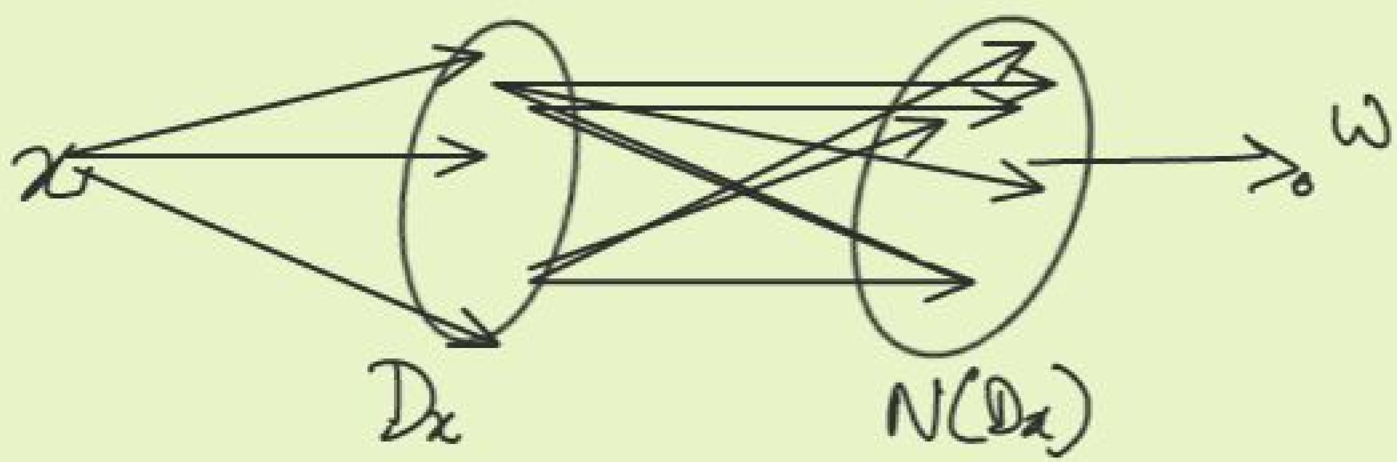
$w \notin N(D_x)$

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{y \mid x \text{ defeats } y\}$

$N(D_x) = \{z \mid \exists y \in D_x \text{ and } y \text{ defeats } z\}$



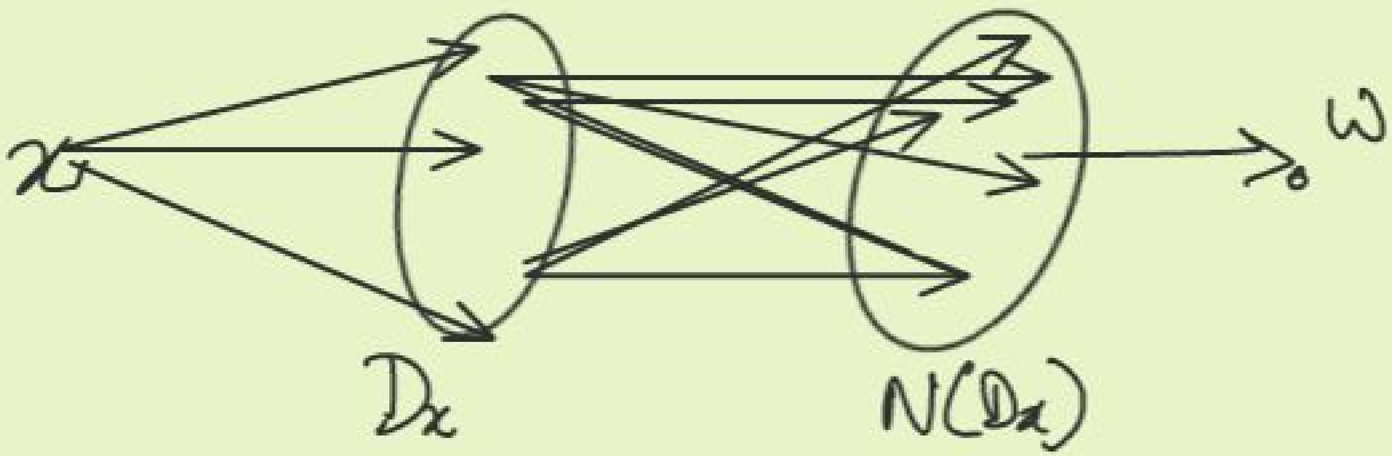

$\because w \notin D_x$, $w$ must defeat $x$.

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{y \mid x \text{ defeats } y\}$

$N(D_x) = \{z \mid \exists y \in D_x \text{ and } y \text{ defeats } z\}$




$\because w \notin D_x$, $w$ must defeat $x$.

$\because w \notin N(D_x)$, $w$ must defeat all vertices in $D_x$

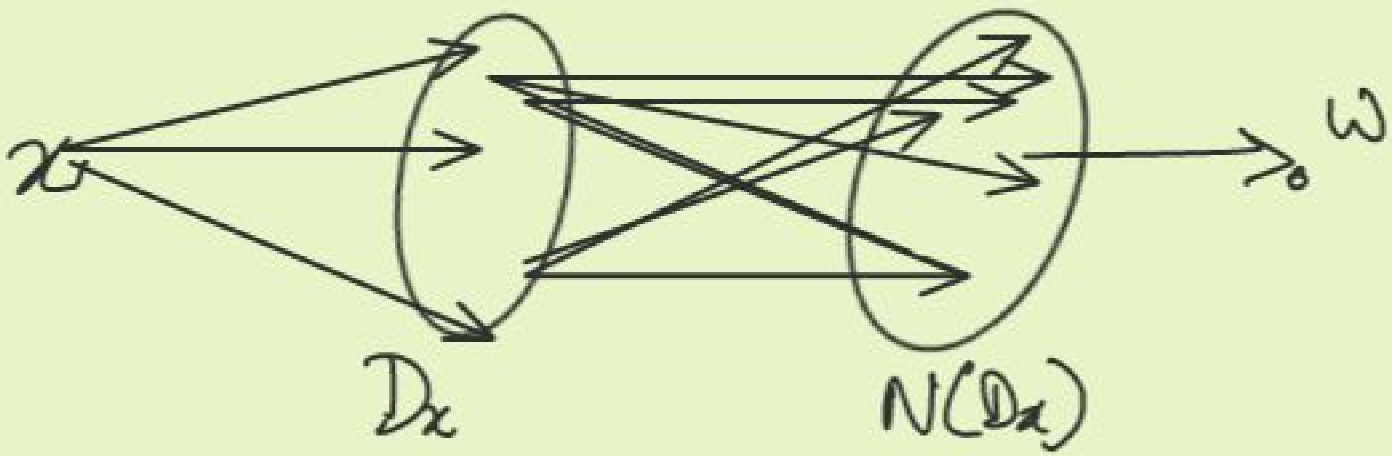
$\implies D_x \cup \{x\} \subseteq D_w$

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{y \mid x \text{ defeats } y\}$

$N(D_x) = \{z \mid \exists y \in D_x \text{ and } y \text{ defeats } z\}$




$\therefore$ $w \notin D_x$, $w$ must defeat $x$.

$\therefore$ $w \notin N(D_x)$, $w$ must defeat all vertices in $D_x$

Every tournament has a king.

<u>proof</u>: Let $x \in V$ be any arbitrary vertex in the tournament.

Let $D_x = \{ y \mid x \text{ defeats } y \}$

$N(D_x) = \{ z \mid \exists y \in D_x \text{ and } y \text{ defeats } z \}$

Let $w \in V$ s.t. $w \notin D_x$, $w \notin N(D_x)$

$$\Downarrow$$

$$D_x \cup \{x\} \subseteq D_w$$

i.e. $|D_x| < |D_w|$.

Every tournament has a king.

<u>proof</u>:

$\therefore \forall x \in V$ s.t. $x$ is not a king

$\exists\, w \in V$ s.t. $w \notin D_x$ and $w \notin N(D_x)$

& $|D_x| < |D_w|$

But as the graph is finite, the process cannot continue forever.

A tournament is said to have a $k$-Winner property if

$\forall U \subseteq V$ s.t. $|U| = k$ $\exists x \in V$ s.t.

$x$ defeats all vertices in $U$.

If $k < \frac{\log n}{100}$ then $\exists$ a tournament on $n$ vertices with the $K$-winner property.

If $k < \frac{\log n}{100}$ then $\exists$ a tournament on $n$ vertices with the $k$-winner property.

<u>proof</u>: pick a random tournament $T$

Fix a subset $U \subseteq V$, $|U| = k$.

$$\Pr_T\left[\begin{array}{c} U \text{ is not defeated by all vertices} \\ \text{in } V \setminus U \end{array}\right] = (1-2^{-k})^{n-k}$$

$$\Pr\left[\begin{array}{c} \exists U : U \text{ is not defeated by all vertices} \\ \text{in } V \setminus U \end{array}\right]$$

$$\leq \binom{n}{k} (1-2^{-k})^{n-k}$$

CS 207

Discrete Structures

Nutan Limaye

15 OCT 2013

Last Class :

- Directed graphs and tournaments
- Existence of a king
- Existence of a global winner.

Today:

- Brief recap of the last calculation
- Dominating sets in a graph.
- Existence of small dominating sets.

A tournament is said to have a k-Winner property if

$\forall U \subseteq V$ s.t. $|U| = k$ $\exists x \in V$ s.t.

$x$ defeats all vertices in $U$.

If $k < \frac{\log n}{100}$ then $\exists$ a tournament on $n$ vertices with the $k$-winner property.

If $k < \frac{\log n}{100}$ then $\exists$ a tournament on $n$ vertices with the $k$-winner property.

<u>proof</u>: pick a random tournament $T$

Fix a subset $U \subseteq V$, $|U| = k$.

$$\Pr_T\left[ U \text{ is not defeated by all vertices in } V \setminus U \right] = \left(1 - 2^{-k}\right)^{n-k}$$

$$\Pr\left[ \exists U : U \text{ is not defeated by all vertices in } V \setminus U \right] \leq \binom{n}{k} \left(1 - 2^{-k}\right)^{n-k}$$

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called a dominating set if

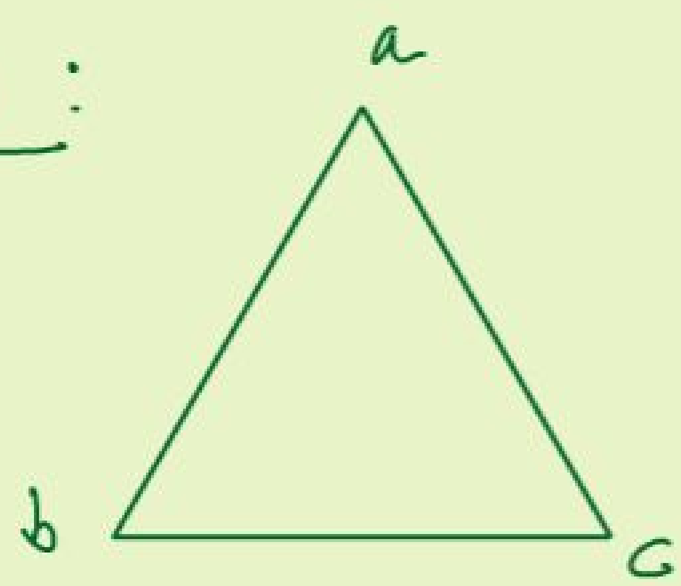
$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } \quad (x, y) \in E \text{ and } y \in U.$$

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called <u>a dominating set</u> if

$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } \quad (x, y) \in E \text{ and } y \in U.$$

<u>Examples</u>:




$\{a\}$

$\{b\}$

$\{a, b\}$

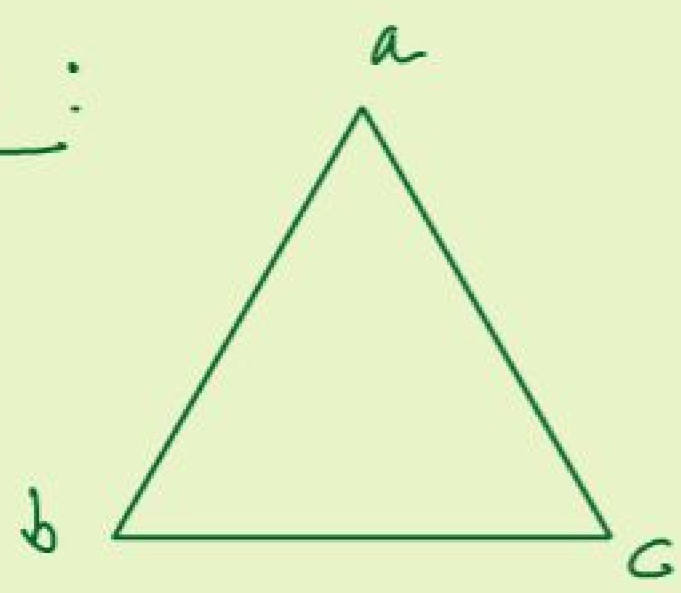
$\{a, b, c\}$

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called <u>a dominating set</u> if

$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } \quad (x, y) \in E \text{ and } y \in U.$$

<u>Examples</u>:




$\{a\}$ ✓

$\{b\}$ ✓

$\{a, b\}$ ✓

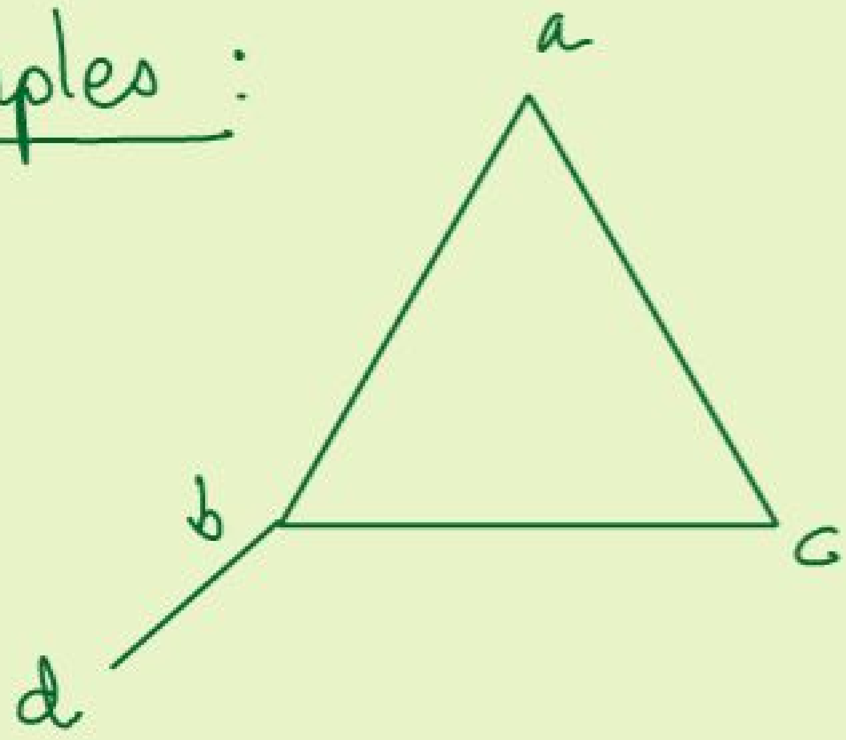
$\{a, b, c\}$ ✓

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called a dominating set if

$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } \quad (x, y) \in E \text{ and } y \in U.$$

Examples:



$\{a\}$

$\{b\}$

$\{a, c\}$

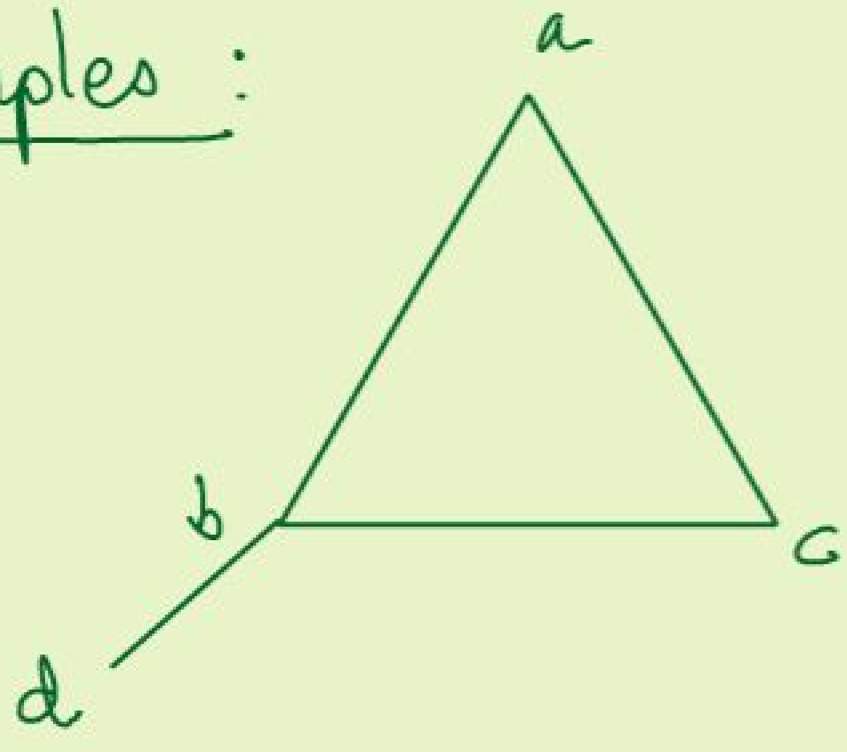
$\{d, c\}$

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called a dominating set if

$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } \quad (x, y) \in E \text{ and } y \in U.$$

Examples:




$\{a\}$ ✗

$\{b\}$ ✓

$\{a, c\}$ ✗

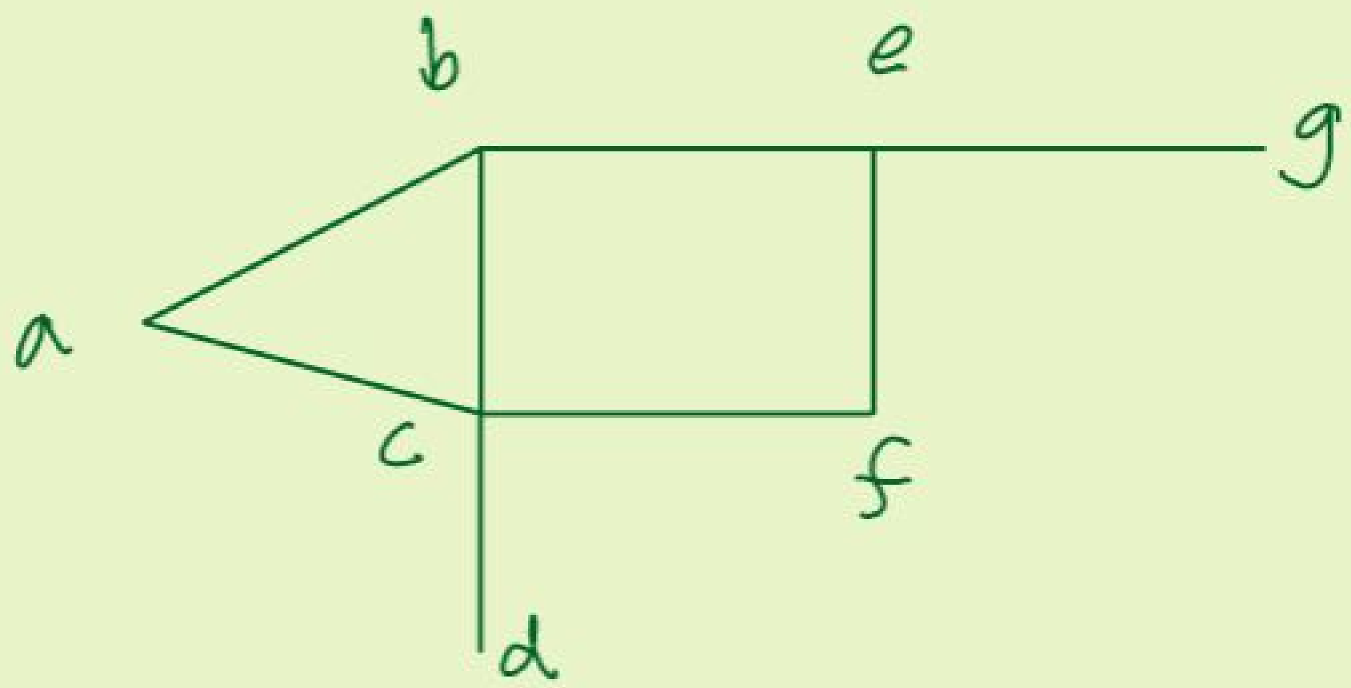
$\{d, c\}$ ✓

Let $G = (V, E)$ be an arbitrary undirected graph.

A subset $U \subseteq V$ is called a dominating set if

$$\forall x \in V \setminus U \quad \exists y \text{ s.t. } (x, y) \in E \text{ and } y \in U.$$

Examples :




What is the size of the smallest dominating set ?

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n\,(1 + \log(\delta+1))}{(\delta+1)}$$

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$, $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n(1 + \log(\delta + 1))}{(\delta + 1)}$$

<u>proof</u>: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

Let $G = (V,E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n\left(1 + \log(\delta+1)\right)}{(\delta+1)}$$

<u>proof</u>: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$$E\left[|X|\right] = np.$$

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n(1+\log(\delta+1))}{(\delta+1)}$$

proof: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$E[|X|] = np$.

Let $Y \subseteq V \setminus X$ s.t. no nbr of $Y$ is in $X$.

for any vertex $u \in V$ $\text{Prob}[u \in Y] = ?$

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n(1 + \log(\delta+1))}{(\delta+1)}$$

proof: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$E[|X|] = np$.

Let $Y \subseteq V \setminus X$ s.t. no nbr of $Y$ is in $X$.

for any vertex $u \in V$ $\text{Prob}[u \in Y] = ?$

$\text{Prob}[u \in Y] = \text{Prob}[u$ & no nbr of $u$ is in $X]$

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n(1+\log(\delta+1))}{(\delta+1)}$$

<u>proof</u>: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$E[|X|] = np$.

Let $Y \subseteq V \setminus X$ s.t. no nbr of $Y$ is in $X$.

for any vertex $u \in V$ $\text{Prob}[u \in Y] = ?$

$\text{Prob}[u \in Y] = \text{Prob}[u$ & no nbr of $u$ is in $X]$

$\leq (1-p)^{\delta+1}$

Let $G=(V,E)$ be a graph with minimum degree $\delta > 1$ & $|V|=n$. Then $G$ has a dominating set of size at most

$$\frac{n(1+\log(\delta+1))}{(\delta+1)}$$

<u>proof</u>: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$\mathbb{E}[|X|] = np$.

Let $Y \subseteq V \setminus X$ s.t. no nbr of $Y$ is in $X$.

for any vertex $u \in V$ $\operatorname{Prob}[u \in Y] = ?$

$\operatorname{Prob}[u \in Y] = \operatorname{Prob}[u \text{ \& no nbr of } u \text{ is in } X]$

$\leq (1-p)^{\delta+1}$. $\Rightarrow$ $\mathbb{E}[|Y|] \leq n(1-p)^{\delta+1}$

Let $G = (V, E)$ be a graph with minimum degree $\delta > 1$ & $|V| = n$. Then $G$ has a dominating set of size at most

$$\frac{n(1+\log(\delta+1))}{(\delta+1)}$$

proof: pick every vertex $x \in V$ w.p. $p$ to obtain $X$.

$$\mathbb{E}[|X|] = np \,;\quad \mathbb{E}[|Y|] \leq n(1-p)^{\delta+1}$$

But note that $X \cup Y$ is a dominating set

$$\mathbb{E}[|X \cup Y|] \leq \mathbb{E}[|X|] + \mathbb{E}[|Y|]$$

$$\leq np + n(1-p)^{\delta+1}.$$

$\therefore \exists$ a dominating set of size $\leq np + n(1-p)^{\delta+1}$.