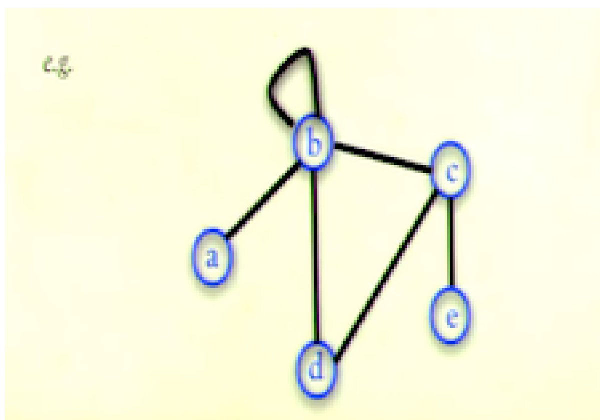
Lecture 7: Graph Theory Review

1 Undirected Graphs

- An undirected **graph** $G = (V, E)$ consists of :
  - A set V of **vertices** (or **nodes**).
  - A set E of **edges** (or **links**) denoting unordered vertex pairs.

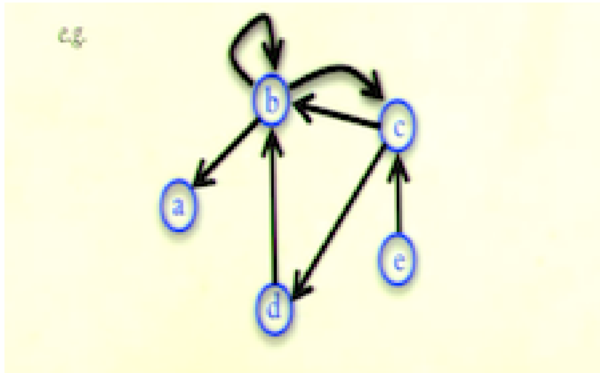


- We set $n = |V|$ to be the cardinality of the vertex set.
- We set $m = |E|$ to be the cardinality of the edge set.

2 Undirected Graphs

- A directed **graph** $G = (V, E)$ consists of :
  - A set V of **vertices**.

- ○ A set A of **arcs** (directed edges) denoting *ordered* vertex pairs.

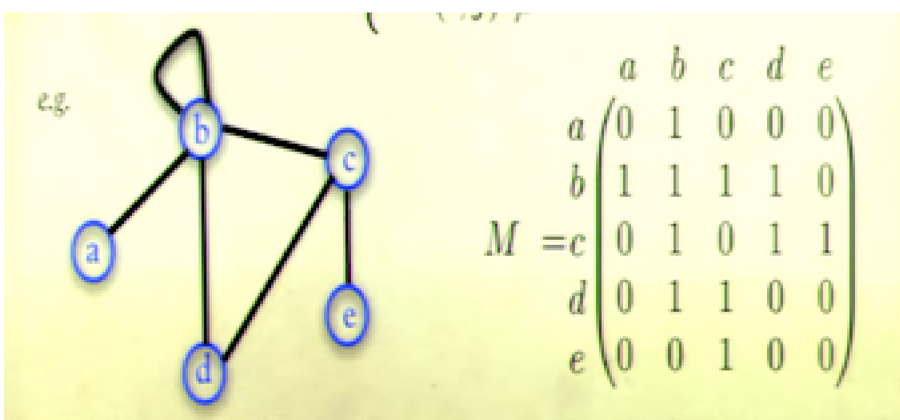


- We set $n = |V|$ to be the cardinality of the vertex set.
- We set $m = |A|$ to be the cardinality of the arc set.

3 Adjacency Matrix (Undirected Graphs)

- For an undirected graph, an **adjacency matrix** M has the properties that:
  - ○ There is a **row** for each vertex.
  - ○ There is a **column** for each vertex.
  - ○ The ij-th **entry** of the matrix is defined by:

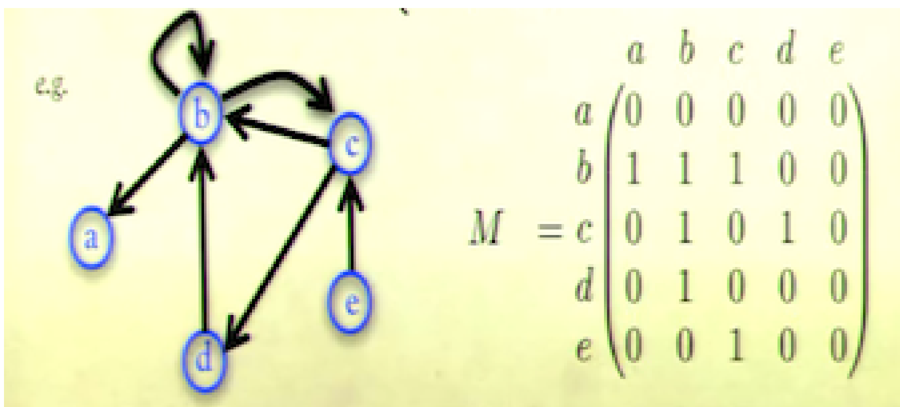
$$M_{ij} = \begin{cases} 1, & (i,j) \in E. \\ 0, & (i,j) \notin E. \end{cases}$$



4 Adjacency Matrix (Directed Graphs)

- For a directed graph, an **adjacency matrix** M has the properties that:
  - There is a **row** for each vertex.
  - There is a **column** for each vertex.
  - The ij-th **entry** of the matrix is defined by:

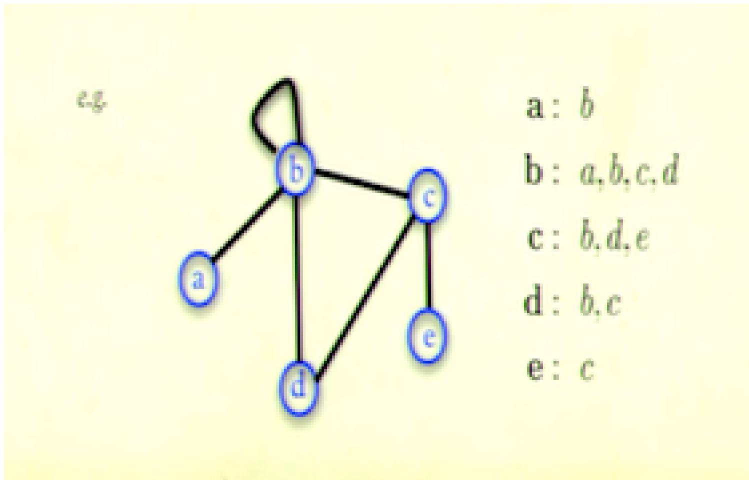
$$M_{ij} = \begin{cases} 1, & (i,j) \in A. \\ 0, & (i,j) \notin A. \end{cases}$$



Note: No longer symmetric as undirected graphs!

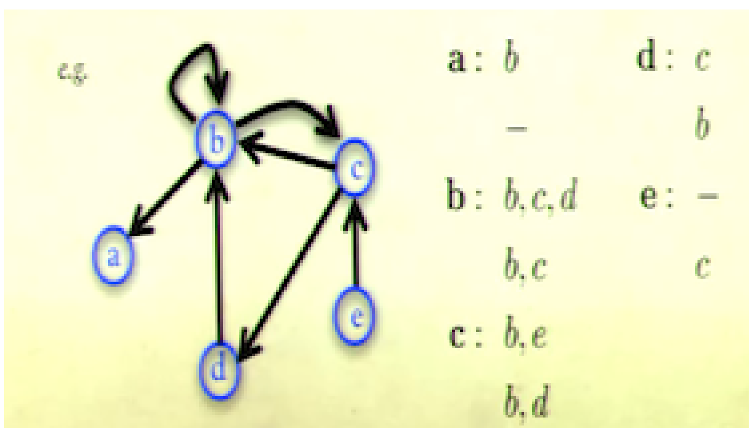
5 Adjacency Lists (Undirected Graphs)

- An undirected graph can also be stored using **adjacency lists**.
  - For each vertex i, we store a list of the **neighbours** of i.
    Neighbours: the set of vertices that i shares an edge with.
  - Equivalently, we store a list of the edges *incident* to each vertex.



6 Adjacency Lists (Directed Graphs)

- A directed graph can also be stored using **adjacency lists**.
  - For each vertex i, we store a list of the **in-neighbours** of i.
  - For each vertex i, we store a list of the **out-neighbours** of i.
    In-neighbours: the set of vertices with arcs that point to i.
    Out-neighbours: the set of vertices that i has arcs pointing to



7 Adjacency Lists versus Adjacency Matrices

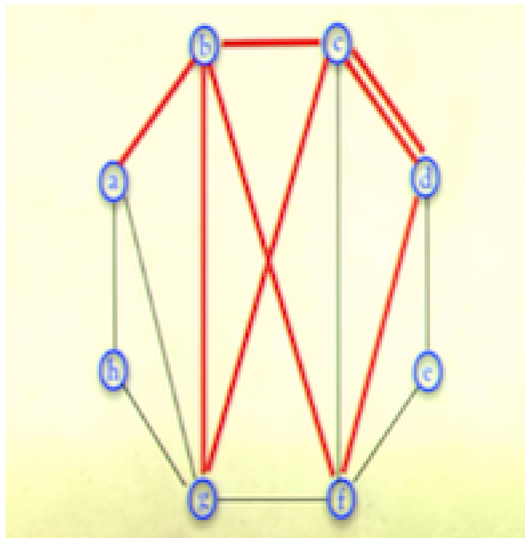
- The main difference is in the amount of **storage** required
  - An *adjacency* matrix requires storing $\Theta(n^2)$ numbers.
  - An *adjacency* list requires storing $\Theta(m)$ numbers.
    - → Each arc will appear twice in a list.
- In any graph $m = O(n^2)$ and often $m \ll n^2$
  - In **sparse graphs** it is much more preferable to use adjacency lists.

8 Graph Application

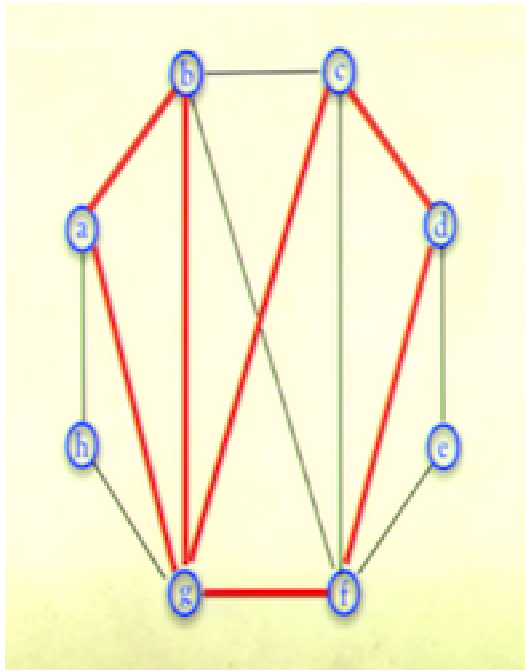
- The number of practical applications for is very large.
- Obviously there are useful in modelling <u>networks</u>.
  - Transportation Networks: e.g. Roads.
  - Social Networks: e.g. Facebook Graph.
  - Information Networks: e.g. the World Wide Web.
  - Financial Networks: e.g. Monetary Flows.
- But, their flexibility allows them to model a plethora of diverse applications:
  - Hierarchy: data structures, linguistics.
  - Similarity: data clustering, biology.
  - Conflict: wavelength allocation, scheduling.
  - Priority: industrial planning, operations research.
  - Structure: chemistry, physics.
  - Time Relations: evolution, migration patterns.

9 Graph Structure

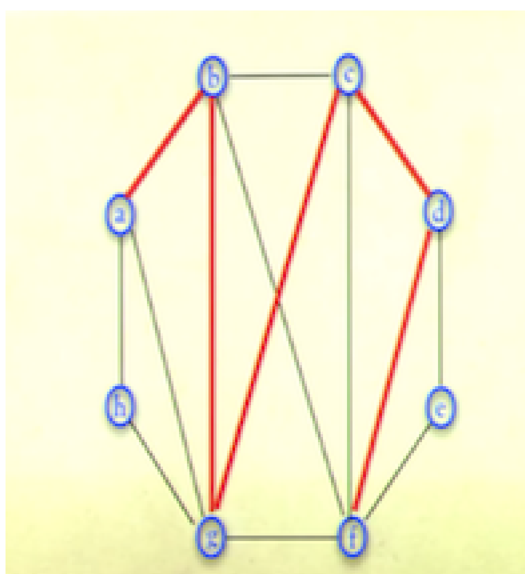
- Walks
  - A **walk** is list of vertices $\{v_0, v_1, ..., v_l\}$ such that $(v_i, v_{i+1}) \in E$ for all $0 \le i < l$.



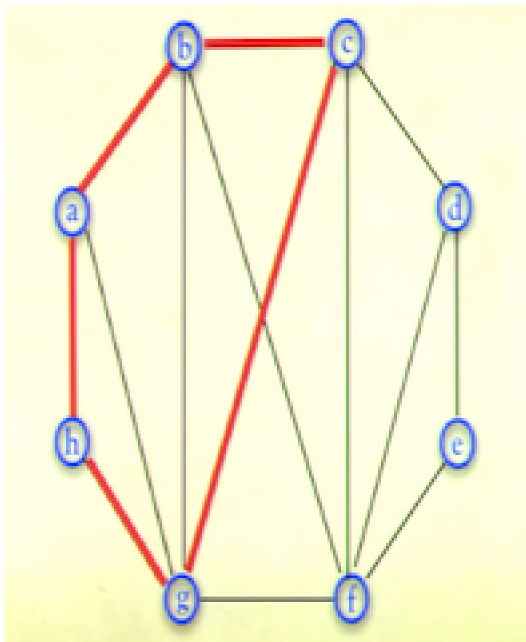
- Circuits
  - A **circuit** is a walk $\{v_0, v_1, ..., v_l\}$ where $v_0 = v_l$.
  - A circuit is a **closed** walk.
  - We are not allowed to use same edge/arc more than once.
  - An **Eulerian Circuit** is a circuit that uses every edge exactly once.



- Paths
  - A **path** is a walk where every vertex is distinct.
  - Revisit is not allowed.

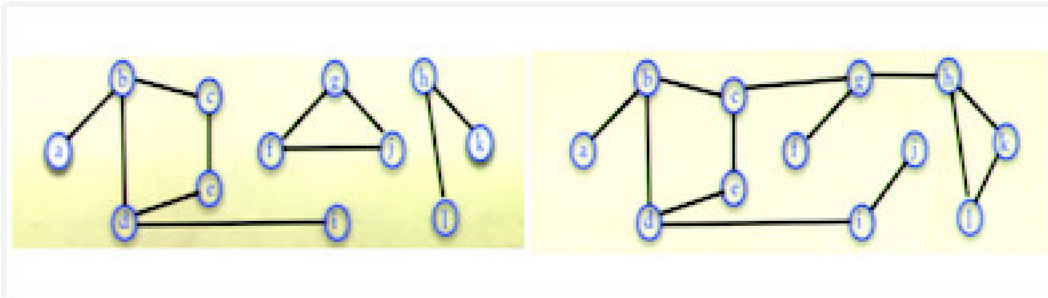


- Cycles
  - A **cycle** is a walk $\{v_0, v_1, ..., v_l\}$ where every vertex is <u>distinct</u> except for the end-vertices $v_0 = v_l$.
  - A cycle is a **closed** path.
  - An **Hamiltonian Cycle** is a cycle that uses every vertex exactly once.

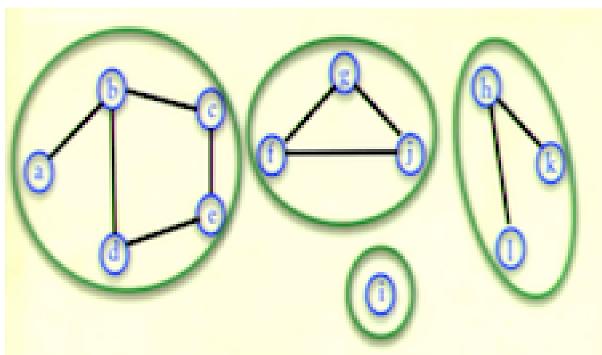


10 Connected Graphs

- A graph is **connected** if for every pair of vertices $u, v \in V$, it is possible to walk from u to v.
- A graph is **disconnected** if there exists a pair of vertices $u, v \in V$, for which there is no possible walk from u to v.

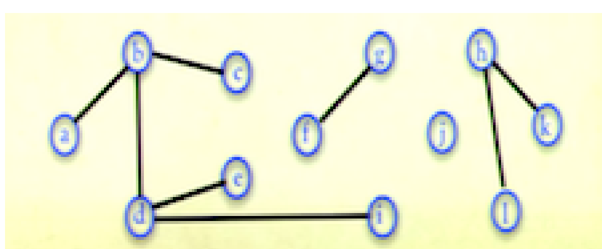


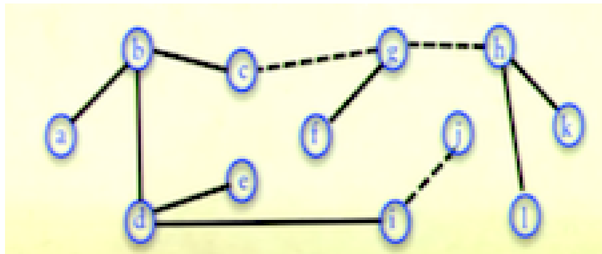
- Graph Components
  - A connected subgraphs are called the **components** of the graph.
  - Thus a connected graph has exactly <u>one</u> component.



11 Trees

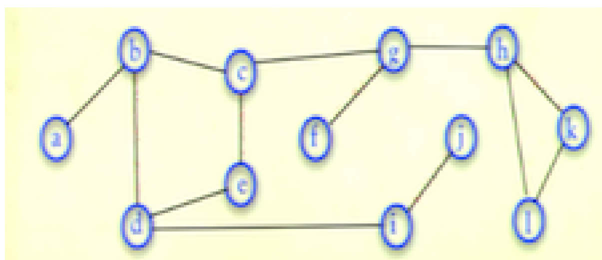
- A **tree** is a connected component with no cycles.
- A **forest** is a graph whose components are all trees.
- A tree is **spanning** if it contains every vertex in the graph.

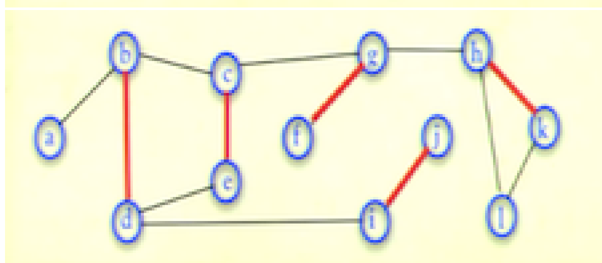




12 Matchings

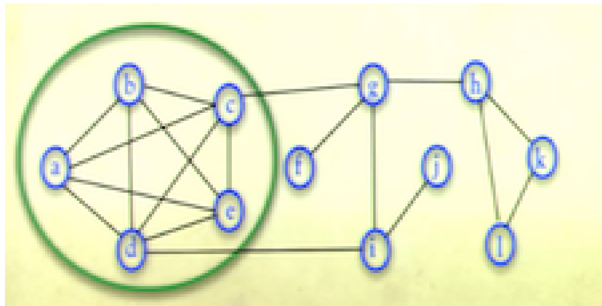
- A **matching** is a set of vertex-disjoint edges.
- Hence, each vertex is incident to at most one edge in the matching.
- A matching is **perfect** if every vertex is incident to an edge in the matching.





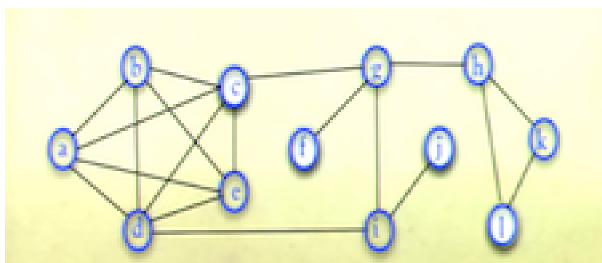
13 Cliques

- A **clique** is a set of pairwise adjacent vertices.



14 Independent Sets

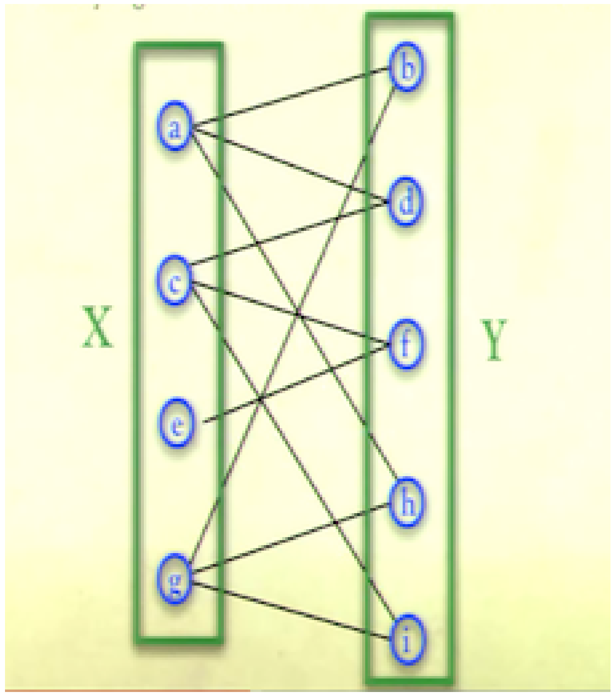
- An **independent set (stable set)** is a set of pairwise non-adjacent vertices.
- If you find a stable set, then there is no conflict. (Assume the graph representing conflicts.)



(c, f, j, l)

15 Bipartite Graphs

- In a **bipartite graph** the vertex set can be partitioned as $V = X \cup Y$ such that every edge has one end-vertex in X and one end-vertex in Y.



Note that X and Y are both independent sets.

16 Some Theorems on Undirected Graphs

- The Handshaking Lemma
  - Let $\tau(v) = \{u : (u, v) \in E\}$ be the set of neighbours of v.
  - The degree, $deg(v)$, of a vertex v is the cardinality of $\tau(v)$.

The Handshaking Lemma. In an undirected graph, there are an even number of vertices with odd degree.
Proof.

- We have:

$$2 \cdot |E| = \sum_{v \in V} deg(v)$$

(Double count the number of pairs (v,e) where e is an edge incident to v.)

$$= \sum_{v \in O} deg(v) + \sum_{v \in \varepsilon} deg(v)$$

(O is the set of vertices with odd degree, and ε is the set of vertices with even degree.)

$$\Rightarrow \sum_{v \in O} deg(v) = 2 \cdot |E| - \sum_{v \in \varepsilon} deg(v)$$

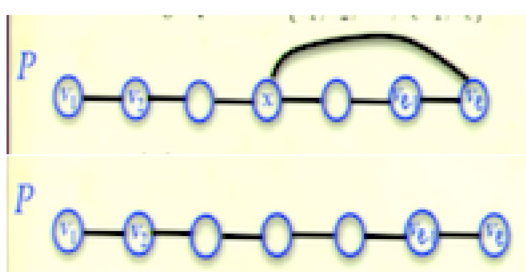
Even = Even - Even

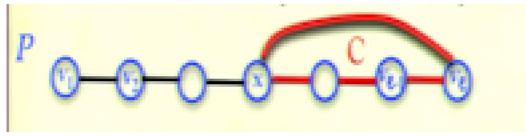
17 Euler's Theorem

- The first result in Graph Theory is the following.
  Theorem. An undirected graph contains an Euler Circuit if and only if every vertex has an even degree.
  Proof. Exercise! (Use induction.)

18 A Theorem on Trees

- Leaves
  - A vertex with degree one in a tree is called a **leaf.**
    Lemma. A tree T with $n \geq 2$ vertices has at least one leaf vertex.
    Proof.
    - → A tree is connected ⇒ There are no vertices with degree 0 as $n \geq 2$.
    - → For a contradiction, assume every vertex has degree at least 2.
    - → Take the **longest path** $P = \{v_1, v_2, ..., v_{l-1}, v_l\}$ in T.
    - → But $deg(v_k) \geq 2$ so v_l has a neighbour $x \neq v_{l-1}$
    - → We must have $x = v_j$ for some $1 \leq j \leq l-2$ otherwise $\{v_1, v_2, ..., v_l, x\}$ is a longer path than P.
    - → But then $C = \{x = v_j, v_{j+1}, ..., v_l, x\}$is a cycle, contradiction.





- The Number of Edges in a Tree
  Theorem. A tree with n vertices has n-1 edges.
  Proof.

  - Let's prove this by induction.

  Base Case:

  - A tree on **one** vertex has **zero** edges.

  Induction Hypothesis:

  - Assume that any tree on n-1 vertices has n-2 edges.

  Induction Step:

  - Take a tree T with $n \geq 2$ vertices.
  - By the previous lemma, this tree contains a **leaf** vertex v.
    $\Rightarrow$ T $\setminus$ {v} is a tree on n-1 vertices.
    $\Rightarrow$ By the induction hypothesis, T $\setminus$ {v} is a tree on n-2 vertices.
    $\Rightarrow$ T is a tree with n-1 edges.

19 Hall's Theorem

- How do we know if a bipartite graph $G = (X \cup Y, E)$ contains a **perfect matching**?
- This is actually easy to test using Hall's Condition.

$$\underline{\text{Hall's Condition:}}\ \forall B \subseteq X,\ |\tau(B)| \geq |B|$$

Hall's Theorem. A bipartite graph, with $|X| = |Y|$, contains a perfect matching if and only if Hall's condition is satisfied.
Proof.
($\Rightarrow$)

- If there is a set $B \subseteq X$ with $|\tau(B)| < |B|$ then the graph **cannot** have a perfect matching.

($\Leftarrow$)

- Suppose Hall's Condition is satisfied: $\forall B \subseteq X$, $|\tau(B)| \geq |B|$
- Take a **maximum cardinality** matching M in the graph.
- If M is perfect we are done.
- So we may assume M is not perfect and there is an **unmatched** vertex b_0 in X.
- As Hall's condition holds we have: $|\tau(\{b_0\})| \geq |\{b_0\}| = 1$
- So b_0 has at least one neighbour s_0
- Let s_0 be matched to b_1 in M
- As Hall's condition holds we have: $|\tau(\{b_0, b_1\})| \geq |\{b_0, b_1\}| = 2$
- So either b_0 or b_1 has a neighbour $s_1 \neq s_0$
- Let s_1 be matched to b_2 in M.
- As Hall's condition holds we have: $|\tau(\{b_0, b_1, b_2\})| \geq |\{b_0, b_1, b_2\}| = 3$
- So either $\{b_0, b_1, b_2\}$ has a neighbour $s_2 \neq \{s_0, s_1\}$
- As Hall's condition holds we have: $|\tau(\{b_0, b_1, b_2, b_3\})| \geq |\{b_0, b_1, b_2, b_3\}| = 4$
- So either $\{b_0, b_1, b_2, b_3\}$ has a neighbour $s_3 \neq \{s_0, s_1, s_2\}$
- ...
- The graph contains a finite number of nodes so this process must terminate.
- The process can only terminate if we reach an unmatched node $s_k \in S$
- Using the edges we have found we can trace back a **path** P from s_k to b_0 that alternates between using non-matching edges and matching edges.
- **Swapping** the matching and non-matching edges gives one extra matching edge.
- Note this is still a <u>valid</u> matching. Firstly, the internal nodes of P are still incident to exactly one matching edge.
- Secondly, its end-nodes, s_k and b_0, were previously unmatched so are now incident to exactly one edge in the new matching.

- This contradicts the fact that M was a maximum cardinality matching.