**A proposition (命题) is a declarative sentence that is either true or false.命题是一个具有确定真值（真或假）的陈述句**

**Simple Proposition(简单命题): cannot be broken into 2 or more propositions不可再分解为更小命题**

**• 2 is irrational.**

**Compound Proposition(复合命题): not simple**

**• 2 is rational and 2 is irrational. 非简单命题**

**Propositional Constant(命题常项): a concrete proposition**

**具体命题实例**

**• Every even integer 𝑛 > 2 is the sum of two primes.**

**Propositional Variable(命题变项): a variable that represents any proposition 表示任意命题的变量**

**• Lowercase letters denote proposition variables: 𝑝, 𝑞, 𝑟, 𝑠, …**

**• Truth value is not determined until it is assigned a concrete proposition**

**Propositional Logic(命题逻辑): the area of logic that deals with propositions 研究命题关系的逻辑分支**

**𝑥 + 1 = 2.不是命题 因为x取不同的值 真假不同**

**Implication (→)蕴含 *p*→*q*表示"若p，则q"。**

**等值（↔）双条件语句*p*↔*q*表示"p当且仅当q"**

**"only if" 的含义："A only if B" 在逻辑上表示 A→B，即“A 仅在 B 成立时成立”。**

**合式公式的类型**

**重言式Tautology：所有真值指派下均为真的公式**

**矛盾式Contradiction：所有真值指派下均为假的公式  
可能式Contingency：既非重言式也非矛盾式（如*p*→¬*p*）  
可满足式Satisfiable：至少存在一个真值指派使其为真**

**逻辑等价Logically Equivalent**

**定义：若公式*A*和*B*在所有真值指派下真值相同，则称*A*与*B*逻辑等价（记作*A*≡*B*）。  
定理：*A*≡*B*当且仅当*A*↔*B*是重言式。**

**证明方法：**

**真值表法**

**替换规则：将公式中的子公式替换为逻辑等价的子公式，结果仍等价。**

**Double Negation Law双重否定律¬(¬𝑃) ≡ 𝑃 1**

**Identity Laws同一律 𝑃 ∧ 𝐓 ≡ 𝑃 2**

**𝑃 ∨ 𝐅 ≡ 𝑃 3**

**Idempotent Laws等幂律**

**𝑃 ∨ 𝑃 ≡ 𝑃 4**

**𝑃 ∧ 𝑃 ≡ 𝑃 5**

**Domination Laws零律**

**𝑃 ∨ 𝐓 ≡ 𝐓 6**

**𝑃 ∧ 𝐅 ≡ 𝐅 7**

**Negation Laws补余律**

**𝑃 ∨ ¬𝑃 ≡ 𝐓 8**

**𝑃 ∧ ¬𝑃 ≡ 𝐅 9**

**Commutative Laws交换律**

**𝑃 ∨ 𝑄 ≡ 𝑄 ∨ 𝑃 10**

**𝑃 ∧ 𝑄 ≡ 𝑄 ∧ 𝑃 11**

**Associative Laws结合律**

**𝑃 ∨（𝑄 ∨ 𝑅）≡（𝑃 ∨ 𝑄）∨ 𝑅 12**

**𝑃 ∧（𝑄 ∧ 𝑅）≡（𝑃 ∧ 𝑄）∧ 𝑅 13**

**Distributive Laws分配律**

**𝑃 ∧（𝑄 ∨ 𝑅）≡（𝑃 ∧ 𝑄）∨（𝑃 ∧ 𝑅） 14**

**𝑃 ∨（𝑄 ∧ 𝑅）≡（𝑃 ∨ 𝑄）∧（𝑃 ∨ 𝑅） 15**

**De Morgan’s Laws摩根律**

**¬（𝑃 ∧ 𝑄）≡ ¬𝑃 ∨ ¬𝑄 16**

**¬（𝑃 ∨ 𝑄）≡ ¬𝑃 ∧ ¬𝑄 17**

**Absorption Laws吸收律**

**𝑃 ∨（𝑃 ∧ 𝑄）≡ 𝑃 18**

**𝑃 ∧（𝑃 ∨ 𝑄）≡ 𝑃 19**

**Laws Involving Implication “→”**

**𝑃 → 𝑄 ≡ ¬𝑃 ∨ 𝑄 20**

**𝑃 → 𝑄 ≡ ¬𝑄 → ¬𝑃 21**

**（𝑃 → 𝑅）∧（𝑄 → 𝑅）≡（𝑃 ∨ 𝑄）→ 𝑅 22**

**𝑃 →（𝑄 → 𝑅）≡（𝑃 ∧ 𝑄）→ 𝑅 23**

**𝑃 →（𝑄 → 𝑅）≡ 𝑄 → (𝑃 → 𝑅) 24**

**Laws Involving Bi-Implication “↔”**

**𝑃 ↔ 𝑄 ≡（𝑃 → 𝑄）∧（𝑄 → 𝑃） 25**

**𝑃 ↔ 𝑄 ≡（¬𝑃 ∨ 𝑄）∧（𝑃 ∨ ¬𝑄） 26**

**𝑃 ↔ 𝑄 ≡（𝑃 ∧ 𝑄）∨（¬𝑃 ∧ ¬𝑄） 27**

**𝑃 ↔ 𝑄 ≡ ¬𝑃 ↔ ¬𝑄 28**

**逻辑等价的集合表征**

**定理：设*A*−1(*T*)为使A为真的真值指派集合，则*A*≡*B*当且仅当*A*−1(*T*)=*B*−1(*T*)  
同理，*A*≡*B*当且仅当*A*−1(*F*)=*B*−1(*F*)**

**重言蕴涵 Tautological Implications**

**在 A 为真的所有情况下，B 都不会为假，即 A→B是一个永真式（重言式）。*A* 为真时，命题公式 B 必然为真**

**定义：设A和B是由命题变元*p*1​,…,*pn*​构成的合式公式**

**A重言蕴涵B指所有使A为真的指派也使B为真。记作*A*⇒*B***

**集合表征：*A*−1(*T*)⊆*B*−1(*T*) 或 *B*−1(*F*)⊆*A*−1(*F*)**

**𝐴 ⇒ 𝐵 iff 𝐴 → 𝐵 is a tautology.**

**𝐴 ⇒ 𝐵 iff 𝐴 ∧ ¬𝐵 is a contradiction.**

**以上四种都是其证明方式。**

**Conjunction(合取) (𝑃) ∧ (𝑄) ⇒ 𝑃 ∧ 𝑄 1**

**Simplification(化简) 𝑃 ∧ 𝑄 ⇒ 𝑃 2**

**Addition(附加) 𝑃 ⇒ 𝑃 ∨ 𝑄 3**

**Modus ponens(假言推理) 𝑃 ∧ 𝑃 → 𝑄 ⇒ 𝑄 4**

**Modus tollens(拒取) ¬𝑄 ∧（𝑃 → 𝑄）⇒ ¬𝑃 5**

**Disjunctive syllogism(析取三段论) ¬𝑃 ∧（𝑃 ∨ 𝑄）⇒ 𝑄 6**

**Hypothetical syllogism(假言三段论)**

**（𝑃 → 𝑄）∧（𝑄 → 𝑅）⇒ (𝑃 → 𝑅) 7**

**Resolution (归结)（𝑃 ∨ 𝑄）∧（¬𝑃 ∨ 𝑅）⇒ 𝑄 ∨ 𝑅 8**

**Rules of inference(推理规则): relatively simple valid argument forms from tautological implications**

**谓词​​：描述主语性质的函数（从个体域到 {*T*,*F*}）**

**n元谓词​​：作用于n个个体的谓词（如 *G*(*x*,*y*): “*x*>*y*”）**

**个体词​​：句子中的对象（如常量 *eπ*，变量 *x*）**

**个体域​​：所有可能个体的集合**

**​​命题函数​​：*P*(*x*1​,…,*xn*​)，赋予具体值后成为命题（如 *P*(*eπ*,*πe*)）。**

**全称量词 ∀​​：**

**∀*xP*(*x*) 表示“对所有 *x*，*P*(*x*) 成立”。**

**当个体域为空时，∀*xP*(*x*) 恒为真。**

**​​存在量词 ∃​​：**

**∃*xP*(*x*) 表示“存在 *x* 使得 *P*(*x*) 成立”。**

**当个体域为空时，∃*xP*(*x*) 恒为假。**

**​​约束变量与自由变量​​：**

**若量词作用于变量（如 ∃*x*(*x*+*y*=1)），则 *x* 受约束，*y* 自由。**

**合式公式的类型**

**普遍有效（logically valid）：在所有解释下为真。**

**例：∀*x*(*P*(*x*)∨¬*P*(*x*))**

**不可满足（unsatisfiable）：在所有解释下为假。**

**例：∃*x*(*P*(*x*)∧¬*P*(*x*))**

**可满足（satisfiable）：存在某个解释使其为真。**

**例：∀*x*(*x*2>0)（当论域为非零实数时为真）**

**​​定理​​：合式公式A普遍有效当且仅当¬*A*不可满足。**

**​​替换规则​​：若A是命题逻辑中的重言式，将其命题变量替换为谓词逻辑中的任意合式公式后，所得公式普遍有效。**

**例：*p*∨¬*p*是重言式，故*P*(*x*)∨¬*P*(*x*)普遍有效。**

**THEOREM: 𝐴 ≡ 𝐵 iff 𝐴 𝐵 is logically valid.**

**THEOREM: 𝐴 ≡ 𝐵 iff 𝐴 → 𝐵 and 𝐵 → 𝐴 are both logically valid.**

**量词的德摩根律**

**¬∀*xP*(*x*)≡∃*x*¬*P*(*x*)**

**¬∃*xP*(*x*)≡∀*x*¬*P*(*x*)**

**量词的分配律**

**∀*x*(*P*(*x*)∧*Q*(*x*))≡∀*xP*(*x*)∧∀*xQ*(*x*)**

**∃*x*(*P*(*x*)∨*Q*(*x*))≡∃*xP*(*x*)∨∃*xQ*(*x*)**

**Universal Instantiation**

**全称量词消去**

**∀ 𝑥 𝑃 𝑥 ⇒ 𝑃(𝑎)**

**𝑎 is any individual in the domain of 𝑥**

**Universal Generalization**

**全称量词引入**

**𝑃(𝑎) ⇒ ∀𝑥 𝑃(𝑥)**

**𝑎 takes any individual in the domain of 𝑥**

**Existential Instantiation**

**存在量词消去**

**∃ 𝑥 𝑃 𝑥 ⇒ 𝑃 𝑎**

**𝑎 is a specific individual in the domain of 𝑥**

**Existential Generalization**

**存在量词引入**

**𝑃(𝑎) ⇒ ∃𝑥 𝑃(𝑥)**

**𝑎 is a specific individual in the domain of 𝑥**

**附加前提规则​​：*P*⇒*A*→*B* 等价于 *P*∧*A*⇒*B*。**

**简单图​​：无自环和多重边的有限图**

**加权图​​：每条边被赋予一个正数的图**

**有向图​​：边由有序顶点对 (*u*,*v*) 表示，从 *u* 指向 *v***

**子图​​：若 *H*=(*W*,*F*) 满足 *W*⊆*V* 且 *F*⊆*E*，则 *H* 是 *G* 的子图**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Edges** | **Multiple Edges Allowed?** | **Loops Allowed?** |
| **Simple graph** | **undirected** | **No** | **No** |
| **Multigraph** | **undirected** | **Yes** | **No** |
| **Pseudograph** | **undirected** | **Yes** | **Yes** |
| **Simple directed graph** | **directed** | **No** | **No** |
| **Directed multigraph** | **directed** | **Yes** | **Yes** |
| **Mixed graph** | **undirected + directed** | **Yes** | **Yes** |

**特殊简单图**

**完全图 *Kn*​​​：每对顶点之间恰有一条边**

**环图 *Cn*​​​：顶点按环形连接**

**轮图 *Wn*​​​：环图中心加一个顶点与所有顶点相连**

**立方体图 *Qn*​​​：顶点为 *n* 维二进制串，边连接汉明距离为1的顶点**

**Let 𝐺 = (𝑉 = {𝑣1, … , 𝑣𝑛}, 𝐸) be an undirected graph. The adjacency matrix of 𝐺 is an 𝑛 × 𝑛 matrix 𝐴 = (𝑎𝑖𝑗), where**

**• 𝑎𝑖𝑗 = 𝐦𝐮𝐥𝐭𝐢𝐩𝐥𝐢𝐜𝐢𝐭𝐲重数of 𝑣𝑖, 𝑣𝑗 when 𝑖 ≠ 𝑗**

**• 𝑎𝑖𝑖 = 1 if ∃ a loop from 𝑣𝑖 to itself; 𝑎𝑖𝑖 = 0, otherwise**

**Let 𝐺 = (𝑉 = {𝑣1, … , 𝑣𝑛}, 𝐸) be a simple directed graph. The adjacency matrix of 𝐺 is an 𝑛 × 𝑛 matrix 𝐴 = (𝑎𝑖𝑗), where**

**𝑎𝑖𝑗 = 1 𝑣𝑖, 𝑣𝑗 ∈ 𝐸**

**0 𝑣𝑖, 𝑣𝑗 ∉ 𝐸**

**邻接矩阵​​：**

**无向图：对称矩阵，*aij*​ 表示边 {*vi*​,*vj*​} 的重数**

**有向图：非对称矩阵，*aij*​ 表示弧 (*vi*​,*vj*​) 的重数**

**无向图​​：**

**顶点 *v* 的度 deg(*v*) 是与 *v* 关联的边数（自环计为2）**

**孤立顶点：度为0；悬挂顶点：度为1**

**握手定理​​：2∣*E*∣=∑*v*∈*V*​deg(*v*)且度为奇数的顶点数为偶数**

**有向图​​：**

**入度 deg−(*v*)：以 *v* 为终点的边数**

**出度 deg+(*v*)：以 *v* 为起点的边数**

**定理​​：∑*v*∈*V*​deg−(*v*)=∑*v*∈*V*​deg+(*v*)=∣*E*∣**

**删除边​​：*G*−*e*=(*V*,*E*∖{*e*})**

**添加边​​：*G*+*e*=(*V*,*E*∪{*e*})**

**边收缩​​：将边 *e*={*u*,*v*} 合并为新顶点 *w*，并调整关联边**

**删除顶点​​：移除顶点及其关联边**

**补图​​：顶点集相同，边集为原图未连接的顶点对**

**定义​​：简单图 *G*1​=(*V*1​,*E*1​) 和 *G*2​=(*V*2​,*E*2​) 同构，若存在双射 *f*:*V*1​→*V*2​，使得 {*u*,*v*}∈*E*1​⇔{*f*(*u*),*f*(*v*)}∈*E*2​。**

**二分图定理：简单图是二分图当且仅当可为每个顶点分配两种不同颜色之一，使得相邻顶点颜色不同。**

**匹配：设 *G*=(*V*,*E*) 为简单图。若 *M*⊆*E* 满足 ∀*e*,*e*′∈*M* 有 *e*∩*e*′=∅，则称M为匹配。若顶点v属于某匹配边，则称v被匹配，否则称未匹配。即在M中存在一一对应**

**最大匹配：边数最多的匹配**

**在二分图 *G*=(*A*∪*B*,*E*) 中，若A中所有顶点均被匹配，则称M为从A到B的完全匹配**

**Hall‘s theorem二分图 *G*=(*X*∪*Y*,*E*) 存在从X到Y的完全匹配当且仅当 ∀*A*⊆*X* 有 ∣*N*(*A*)∣≥∣*A*∣。**

**连通性：若无向图G中任意两不同顶点间存在路径，则称G连通。**

**连通分支：图 *G*=(*V*,*E*) 的连通分支是指不被其他连通子图真包含的最大连通子图。**

**顶点v称为割点，若 *G*−*v* 比G有更多连通分支**

**边e称为桥，若 *G*−*e* 比G有更多连通分支**

**顶点连通度：若无向连通图G无割点，则称不可分。**

**定义：设 G=(V,E) 为连通简单图。**

**点割集：使 *G*−*V*′ 不连通的子集 *V*′⊆*V***

**点连通度 *κ*(*G*)：使G不连通或变为 *K*1​ 需删除的最少顶点数**

**若G不连通，*κ*(*G*)=0**

**若 *G*=*Kn*​，*κ*(*G*)=*n*−1**

**否则为最小点割集大小**

**对于n阶简单图G：**

**0 ≤ κ(G) ≤ n-1**

**κ(G)=0 ⇔ G不连通或G=K₁**

**κ(G)=n-1 ⇔ G=Kₙ (n≥2)**

**反证法：若G≠Kₙ，存在不相邻顶点u,v，删除V-{u,v}可使G不连通，故κ(G)≤n-2**

**连通性关系：对任意简单图G，有κ(G) ≤ λ(G) ≤ δ(G)（δ(G)为最小顶点度数）**

**有向图的连通性**

**强连通​​：任意两顶点u≠v间存在双向路径**

**弱连通​​：忽略边方向后无向图连通**

**路径与同构  
长度k≥3的简单回路存在性是简单图的同构不变量。**

**图G的邻接矩阵A的(i,j)元素在Aʳ中表示vᵢ到vⱼ的长度为r的路径数。**

**连通多重图G有欧拉回路 ⇔ 所有顶点度数为偶数**

**证明：通过构造极大路径证明闭合性**

**偶数证明闭合 如果子图不是完整图 说明有支路从支路出发走一次发现不是最长路径 矛盾 即证**

**连通多重图G有欧拉路径（非回路） ⇔ 恰有两个顶点度数为奇数**

**Hierholzer算法找欧拉回路 找一个回路先 然后剩下的顶点再找回路 全部找完之后 相同边消掉 形成大回路**

**哈密顿路径与回路**

**​​哈密顿路径​​：经过图中每个顶点恰好一次的一条简单路径。**

**必要条件​​：**

**若 *G* 中存在度为1的顶点，则 *G* 不存在哈密顿回路。**

**若 *G* 中存在度为2的顶点，则哈密顿回路必须经过该顶点的两条边。**

**充分条件​​：**

**奥尔定理​​：设 *G*=(*V*,*E*) 为阶数 *n*≥3 的简单图。若对于所有不相邻的顶点对 {*u*,*v*} 满足 deg(*u*)+deg(*v*)≥*n*，则 *G* 存在哈密顿回路。**

**狄拉克定理​​：设 *G*=(*V*,*E*) 为阶数 *n*≥3 的简单图。若每个顶点 *u*∈*V* 满足 deg(*u*)≥*n*/2，则 *G* 存在哈密顿回路。**

**此为奥尔定理的推论：∀*u*∈*V*,deg(*u*)≥*n*/2⇒∀*u*,*v*∈*V*,deg(*u*)+deg(*v*)≥*n*。**

**迪杰斯特拉算法**

**​​目标​​：通过迭代找到从 *a* 到 *z* 的最短路径长度。**

**每次迭代向已标记集合中添加一个顶点。**

**每次迭代进行标签更新：顶点 *w* 的标签为从 *a* 到 *w* 且仅经过已标记顶点的最短路径长度。**

**每次选择标签最小的未标记顶点加入集合。**

**平面图​​：若图 *G*=(*V*,*E*) 可以在平面上绘制且无边交叉​。**

**边交叉​​：除端点外的交点。**

**平面表示​​：无交叉的绘制方式；​​非平面图​​：无法无交叉绘制。**

**​​示例​​：*K*1​,*K*2​,*K*3​,*K*4​ 是平面图。**

***Cn*​（*n*≥3）、*Wn*​（*n*≥3）是平面图。**

**若尔当曲线定理​​：任何简单闭合平面曲线 Γ 将平面分为有界内部区域和无界外部区域。连接两区域的平面曲线必与 Γ 相交。**

**​​示例​​：二分图 *K*3,3​ 非平面。**

**在 *K*3,3​ 中选择简单回路 *v*1​,*v*5​,*v*2​,*v*4​,*v*1​**

**若 *K*3,3​ 为平面图，该回路形成简单闭合平面曲线**

**添加 *v*3​,*v*6​ 及其关联边时，由若尔当曲线定理必产生交叉。**

**设 *G*=(*V*,*E*) 为平面图。其边将平面划分为若干​​区域​​（面）。**

**外部面​​：无限区域；​​内部面​​：其余区域**

**边界​​：区域的边集**

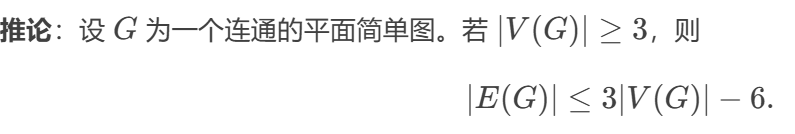
**度数​​：边界边数。共享边对两区域各贡献1，独占边对单区域贡献2。**

**欧拉公式 注意！平面简单图 简单图里无重边和自环！**

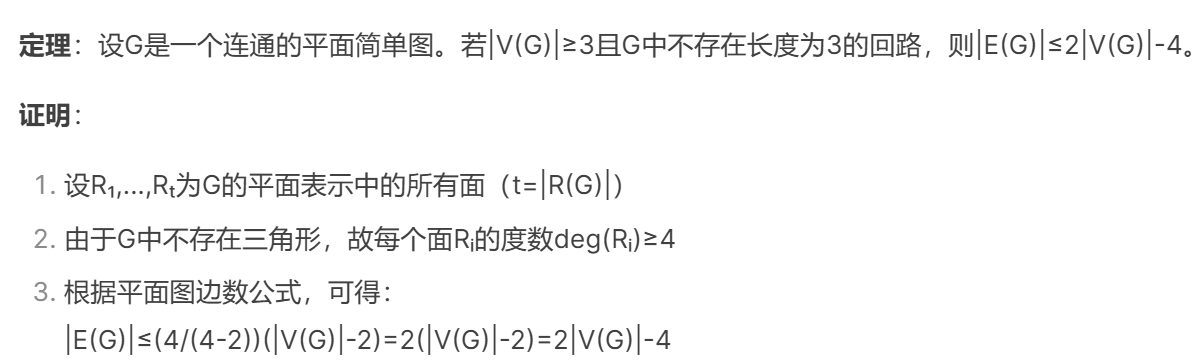
**​​定理​​：设 *G*=(*V*,*E*) 为连通平面简单图，含 *e* 条边、*v* 个顶点。其平面表示有 *r* 个区域，则 *r*=*e*−*v*+2。**

**​​推广​​：若 *G* 含 *p* 个连通分支，则 ∣*V*(*G*)∣−∣*E*(*G*)∣+∣*R*(*G*)∣=*p*+1。**

****

****

****

****

**设G=(V,E)为一图，且{u,v}∈E：  
• 初等细分操作：将边{u,v}替换为两条新边{u,w}和{v,w}，其中w为新添加的顶点，即G'=(V∪{w}, E-{{u,v}}∪{{u,w},{v,w}})  
• 两个图称为同胚的，当且仅当它们可以通过对同一个图进行若干次初等细分操作得到**

**库拉托夫斯基定理​​：图 *G* 非平面当且仅当其含有与 *K*3,3​ 或 *K*5​ 同胚的子图。**

**对偶图​​：平面图 *G* 的​​对偶图​​ *G*∗ 以 *G* 的面为顶点，共享边的面间连边。**

**​​*G*∗ 必为连通平面图**

**对偶图依赖平面表示（不同表示可能非同构）**

**连通时满足 *v*∗=*f*, *e*∗=*e*, *f*∗=*v* 且 *G*​∗∗​=*G***

**自对偶图​​：如轮图 *Wn*​，满足 *e*=2*v*−2**

**图着色：设 *G*=(*V*,*E*) 为简单图。G的k-着色指映射 *f*:*V*→[*k*] ，满足当 {*u*,*v*}∈*E* 时 *f*(*u*)=*f*(*v*) 。**

**色数 (*χ*(*G*)) ：使G存在k着色的最小k值。**

**由于存在长度为3的环路a; b; c，色数至少为3**

**定理：设 *G*=(*V*,*E*) 为简单图。**

**· 1≤*χ*(*G*)≤∣*V*∣**

***χ*(*G*)=1 当且仅当*E*=∅**

***χ*(*G*)=2 当且仅当G是二分图且 ∣*E*∣≥1 。**

***χ*(*Kn*​)=*n* （对任意 *n*≥1 ）。**

**· 若G包含与 *Kn*​ 同构的子图，则 *χ*(*G*)≥*n***

**· *χ*(*Cn*​)=2 （当 2∣*n*）；*χ*(*Cn*​)=3 （当 2∣(*n*−1)；*n*≥3）**

***χ*(*G*)≤Δ(*G*)+1 ，其中 Δ(*G*)=max{deg(*v*):*v*∈*V*} 。**

**定理（四色定理）简单平面图的色数不超过4。**

**定理（五色定理）平面简单图的色数不超过5。**

**树是无简单回路的连通无向图。连通且无简单回路**

**森林是各连通分量均为树的图。**

**(1) 连通**

**(2) 无简单回路**

**(3) (*n*−1) 条边（n为顶点数）**

**若每个内节点最多有m个子节点，则称该有根树为m叉树。**

**若每个内节点恰好有m个子节点，则称该有根树为满m叉树**

**m=2时的m叉树称为二叉树。此时内节点的两个子节点分别称为左子节点和右子节点。以顶点左（右）子节点为根的子树称为该顶点的左（右）子树。**

**满m叉树有i个内节点时，包含 *n*=*mi*+1 个顶点。**

**每生m个小孩 内部顶点+1 叶子就+m**

**满m叉树的性质**

**n个顶点时，有 *i*=(*n*−1)/*m* 个内节点和 ℓ=((*m*−1)*n*+1)/*m* 个叶节点，**

**i个内节点时，有 *n*=*mi*+1 个顶点和 ℓ=(*m*−1)*i*+1 个叶节点，**

**ℓ 个叶节点时，有 *n*=(*m*ℓ−1)/(*m*−1) 个顶点和 *i*=(ℓ−1)/(*m*−1) 个内节点。**