第7章 图论

7-6 树、有向树和有序树

北航计算机学院：李建欣

Tel：82339274（G506）

E-mail：lijx@buaa.edu.cn

http://act.buaa.edu.cn/lijx

主要内容

1. 图论的基本概念
2. 子图和图的运算
3. 路径、回路和连通图
4. 欧拉图和哈密顿图
5. 图的矩阵表示
6. 树、有向树和有序树

§7.6 树

树、有向树、有序树

目的：树的六种定义，了解分支、森林、生成树、生成森林、最小生成树、枝、弦、基本回路、有向树、有向森林、二叉树、最优二叉树、有序树、有序森林、定位二元有序树等概念和性质；掌握求最小生成树、最优二叉树的算法、定位二元有序树和有序森林的双射关系，以及有关的证明方法；

重点：树的六种定义，各种概念、算法及基本的证明思路；

难点：通过树的六种定义方式如何发现树的各种性质，大量相关知识点在证明种的综合运用。

概念图谱

无向图G

什么是树？

树 —G去掉连通约束→ 森林

6个条件(连通;非循环;无自圈;v和v’有唯一路径；G-e非连通;n(E)=n-1)

是否存在？

生成森林：任意G都存在

生成树：**连通**图G

如何找？

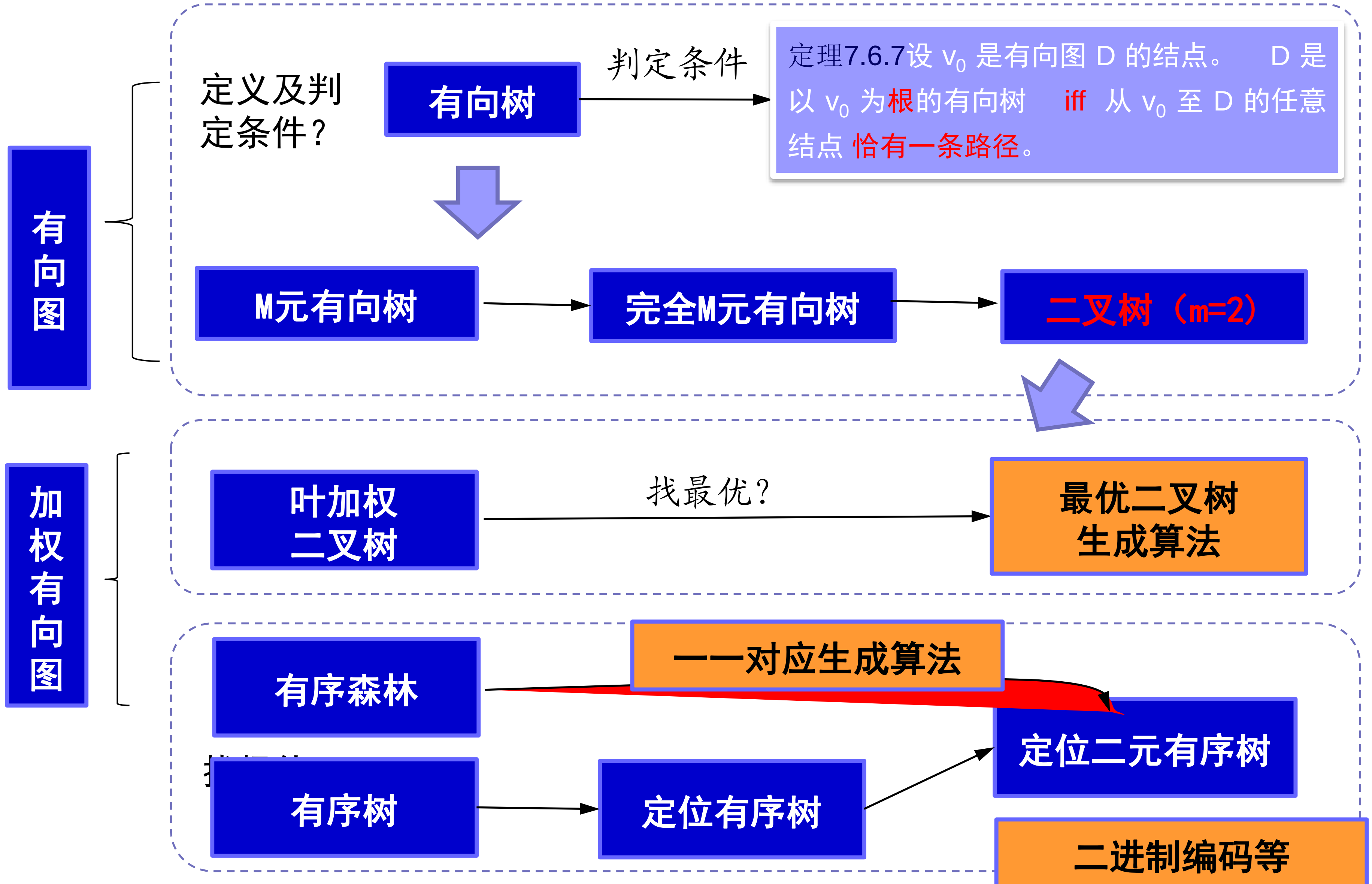
N阶连通无向图 —避圈法 / 破圈法→ 树

加权图

找极值？

最小生成树算法（MST）

概念图谱


有向图
定义及判定条件？
有向树
判定条件
定理7.6.7设 v_0 是有向图 D 的结点。 D 是以 v_0 为根的有向树 iff 从 v_0 至 D 的任意结点 恰有一条路径。
M元有向树
完全M元有向树
二叉树（m=2）
加权有向图
叶加权二叉树
找最优？
最优二叉树生成算法
有序森林
一一对应生成算法
定位二元有序树
有序树
定位有序树
二进制编码等

树

非循环的连通无向图称为树。

平凡树：只有一个定点的无向图

树叶：树T中，次数为一的顶点称为树叶

分支顶点：树T中，次数大于1的顶点称为分支顶点

定理7. 6. 1 树定义的等价条件

设 G ＝〈V ， E ，Ψ〉是 n 阶无向图。可以下六个条件等价:

i) G 是连通的，又是非循环的。

ii) G 没有自圈，并且对于 G 的任意两结点 v 和 v′，在 G 中存在唯一的一条从 v 至 v′的基本路径。

iii) G 是连通的，如果 v 和 v′是 G 的两结点，e 不是 G 的边，当令 Ψ′= {〈e，{v，v′}〉}时，则 G + $\{e\}_{\Psi'}$有唯一的一条回路。

iv) G 是连通的，并且对于 G 的任意边 e，G – e 非连通。

v) G是连通的，并且有 n – 1 条边。

vi) G是非循环的，并且有 n – 1 条边。

证明参见 P145 – 146。

i) $\Rightarrow$ ii)

i) G 是连通的，又是非循环的。

ii) G 没有自圈，并且对于 G 的任意两结点 v 和 v′，在 G 中存在唯一的一条从 v 至 v′的基本路径。

因为G是非循环的，所以无自圈。

若 $v_1 v' \in V$ ，则有定理7.3.1和G为连通图知道，必有从v到v'的基本路径。

假如从v到v'的基本路径不唯一，不妨设$v_0 e_1\ v_1 \dots v_{p-1} e_p v_p$和$v_0' e_1' v_1' \dots v_{q-1}' e_q' v_q'$(其中，$v_0 = v = v_0'$且$v_p = v' = v_q'$)为两条不同的基本路径。设$G_1$是G的以$\{v_0, \dots, v_p\}$为结点集合且以$\{e_1\ , \dots, e_p\}$为边集合的子图，$G_2$是$G$的以$\{v_0', \dots, v_q'\}$为结点集合且以$\{e_1', \dots, e_q'\}$为边集合的子图。

任取$e \notin E$，当另$\Psi' = \{< e, \{v, v'\} >\}$时，$G_1 + \{e\}_{\Psi'}$和$G_2 + \{e\}_{\Psi'}$显然都是回路。

因此$G' = (G_1 + \{e\}_{\Psi'}) \oplus = (G_2 + \{e\}_{\Psi'})$是欧拉图和G的子图，且不是零图，所以$G'$必有分平凡分支$G''$。对$G''$的每个结点u显然皆有$d_{G''}(u) > 1$，根据定理7.3.9和$G''$为$G$ 的子图知道，G 不是非循环图。这与G是非循环的矛盾。

定理7. 6. 1

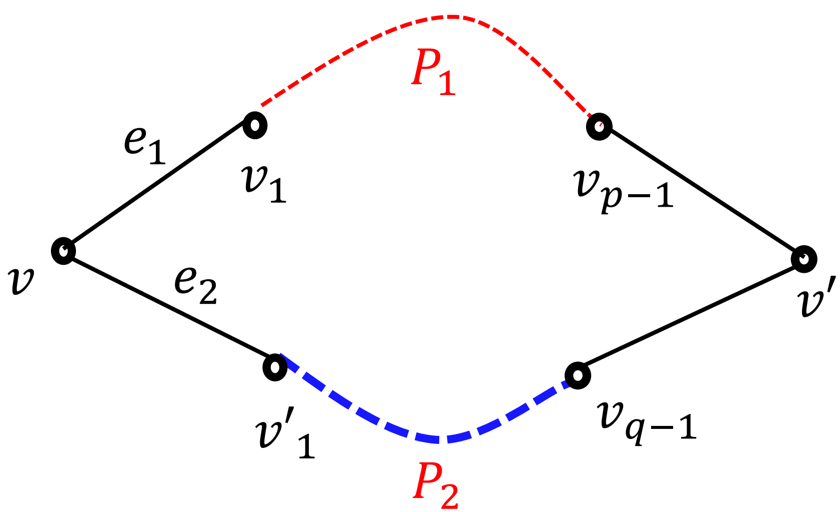
证明：a)**自圈**：因为G是非循环的，所有G没有自圈；

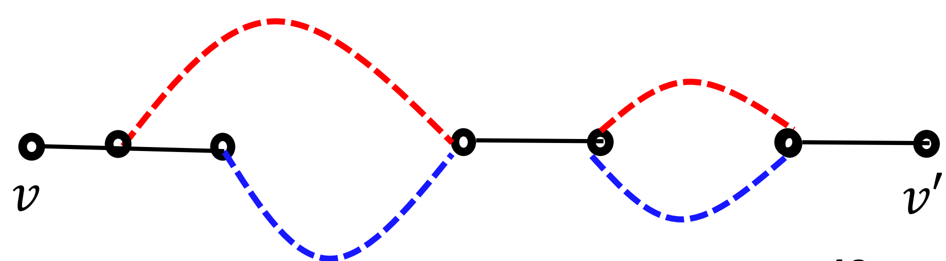
b)**存在性证明**：若v, v′∈V, 则由定理7.3.1和G是连通的，存在从 v 至 v′的基本路径。

c)**路径唯一性证明**：假设路径不唯一，设存在两条路径。证明欧拉图

设G_1是G的以$\{v_0,\ldots,v_p\}$为结点e_1，…，$e_p\}$为边子图，G_2是G的以$\{v'_0,\ldots,v'_q\}$为结点集合且以$\{e'_1,\ldots,e'_q\}$为边集合的子图。

任取$e\notin E$，当另$\Psi'=\{<e,\{v,v'\}>\}$时，$G_1+\{e\}_{\Psi'}$和$G_2+\{e\}_{\Psi'}$显然都是回路。因此$G'=(G_1+\{e\}_{\Psi'})\oplus=(G_2+\{e\}_{\Psi'})$是欧拉图和G的子图





定理7.6.1　ii) ⇒ iii)

ii) G 没有自圈，并且对于 G 的任意两结点 v 和 v′，在 G 中存在唯一的一条从 v 至 v′的基本路径。

iii) G 是连通的，如果 v 和 v′是 G 的两结点，e 不是 G 的边，当令Ψ′= {〈e，{v, v ′}〉}时，则 G+$\{e\}_{\Psi'}$有唯一的一条回路。

若v,v′∈V,则由ii)知道，必有从v到v′的基本路径。因此G 必为连通的。而且对任意e∉E,当令Ψ′={<e,{ v,v′}>}时，$G+\{e\}_{\Psi'}$中必有回路。假如$G+\{e\}_{\Psi'}$中的回路不唯一，不妨设C_1和C_2为它的两个不同回路，显然e 是C_1和C_2的公共边。从而知道: $C_1 \oplus C_2$ 是G 的子图且不为零图，根据定理7.4.5,$C_1 \oplus C_2$ 还是欧拉图。

因此$C_1 \oplus C_2$ 的非平凡分支G ′必是欧拉闭路，故而对G ′中每个节点μ皆有$d_G(u) > 1$,根据定理7.3.9,G必有回路C, 对C 中任意两个节点μ和μ′，必有两条不同的从μ到μ′ 的基本路径，这与条件ii)相矛盾。

定理7.6.1

ii) $\Rightarrow$ iii)

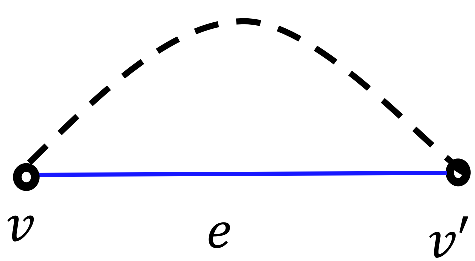
证明： a)**连通性：**存在基本路径，G是连通的；

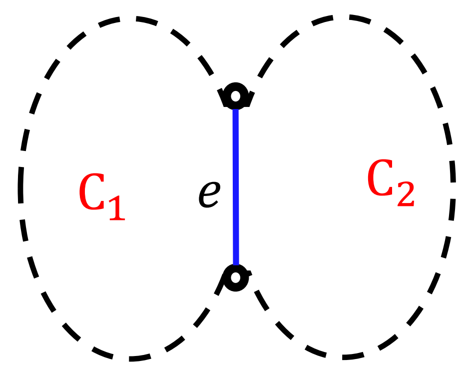
b)**有回路**：若v, v′$\in$V, 则由定理7.3.1和G是连通的，存在从 v 至 v′的基本路径。加上e*存在回路*

c)**回路唯一性**：假设回路不唯一，存在两个回路C_1，C_2。

构造$C_1 \oplus C_2$，证明其为欧拉图，且不包括e（为什么？）.

则原图中有回路$\Leftrightarrow$两条基本路径（与前提矛盾）



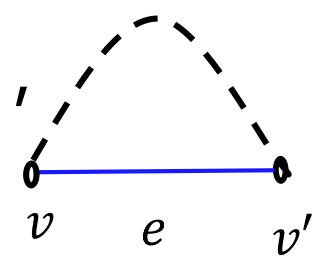


iii) G 是连通的，如果 v 和 v′是 G 的两结点，e 不是 G 的边，当令Ψ′= {〈e，{v，v ′}〉}时，则 G＋{e}$_{\Psi'}$有唯一的一条回路。

iv)G 是连通的，并且对于 G 的任意边 e，G－e 非连通。

iii) ⟹iv) 用反证法

假定iv)不成立，则由G为连通的可知，必有 $e \in E$ 使 $G - e$ 仍是连通的。设 $\Psi(e) = \{v, v'\}$，则G中必有两条不同的从v到v'的基本路径，从而由上面的论证可知，G必有回路。



这时任取$e' \notin E$及u $\in V$，当令$\Psi' = \{< e', \{u\} >\}$时，$G + \{e'\}_{\psi'}$显然有两个不同的回路，这与条件iii）矛盾。

iv)G 是连通的，并且对于 G 的任意边 e，G – e 非连通。

v)G是连通的，并且有 n – 1 条边。

iv)$\Longrightarrow$ v) 显然G是连通的简单图。下面用关于n的第二归纳法证明$n(E) = n - 1$.

a)当$n = 1$时，G显然没有边，即$n(E) = 0$

b)假定对任意的$k \geq 2$，当n<k时皆有$n(E) = n - 1$;

c)假定当$n = k$时有$n(E) = m$.任取$e \in E$，由$G - e$是非连通图可知，$G - e$恰有两个分支G_1与G_2，设$G_i(i = 1,2)$有n_i个节点和m_i条边，根据归纳假设，必有$m_1 = n_1 - 1, m_2 = n_2 - 1$，从而即得到$m = m_1 + m_2 + 1 = n_1 + n_2 - 1$,即$n(E) = m = k - 1$

iv)G是连通的，并且有 n – 1 条边。

v)G是非循环的，并且有 n – 1 条边。

vi) $\Rightarrow$ v)

只须用关于n的归纳法证明G是非循环图即可。

当n=1时，G为平凡图，故为非循环。

若n=k+1，则由$n(E)=n-1=k$得$\sum_{v\in V} d_G(v)=2n(E)=2k$。

但由G为连通的知道，对每个$v\in V$皆有$d_G(v)\geq 1$，所以必有$v'\in V$使$d_G(v')=1$.这时$G-v'$显然是连通的，

阶为n-1=k且边数为$n(E)-1=\mathrm{k}-1$，根据归纳假设，$G-v'$必是非循环的，因此G也必是非循环的。

vi) $\Rightarrow$ i) **显然只须证明G是连通的即可。**

假定G有k个分支$G_1, \ldots, G_k$，设$G_{\mathrm{i}}(1 \le i \le k)$有$n_i$个结点和$m_i$条边 。因为每个$G_{\mathrm{i}}(1 \le i \le k)$都是非循环且为连通的，所以由前面的论证知道必有i) $\Rightarrow$ v)，因此$m_i = n_i - 1(1 \le i \le k)$，从而得到$n - 1 = \sum_{i=1}^{k} m_i = \sum_{i=1}^{k}(n_i - 1) = n - k$，即$k = 1$，这表明G必是连通的

$\Rightarrow$

练习：条件证明

三个基本条件：

i. G是连通的，
ii. G是非循环的，
iii. 有 n – 1 条边。

给定其中两个条件，证明第三个条件

例如：非循环且连通，证明边n-1

- 证明：**施归纳于n**：
  - 当n =1时，由G**非循环**可知：G没有自圈，即G有0条边，故命题为真。
  - 假设对任意k≥1, 当n =k时命题为真。
  - 当n = k+1时：因G为**连通的**，故任意结点v的度数$d_G(v) \geq 1$。
  - 若G中任意结点v的度数$d_G(v) \geq 2$，则G中必存在回路，这与G为**非循环**的条件**矛盾**！因此，G中必有**结点**v_1的度数$d_G(v_1)=1$。
  - 显然，k 阶无向图$G - v_1$连通且非循环，由**归纳假设**$G - v_1$有$k - 1$ 条边。设与v_1相邻的结点为v_2，v_1与v_2的连接边为e，G可由$G - v_1$添加结点v_1与连接边e得到，所以G有k条边，即n =k+1时命题亦为真。
  - 综上所述，命题为真。

定理7.6.2

阶大于 1 的树至少有两个端点。

森林

树是非循环的连通无向图，如果去掉对连通性的要求，

就得到森林的概念。

每个分支都是树的无向图称为**森林**。

定理7.6.3

如果森林 F 有 n 个结点，m 条边和 k 个分支，则 $m = n - k$。

证明：n 个顶点的树有 n-1 条边，设每个分支有V_i个顶点，则：$V_1+V_2+\ldots+V_k = n$;

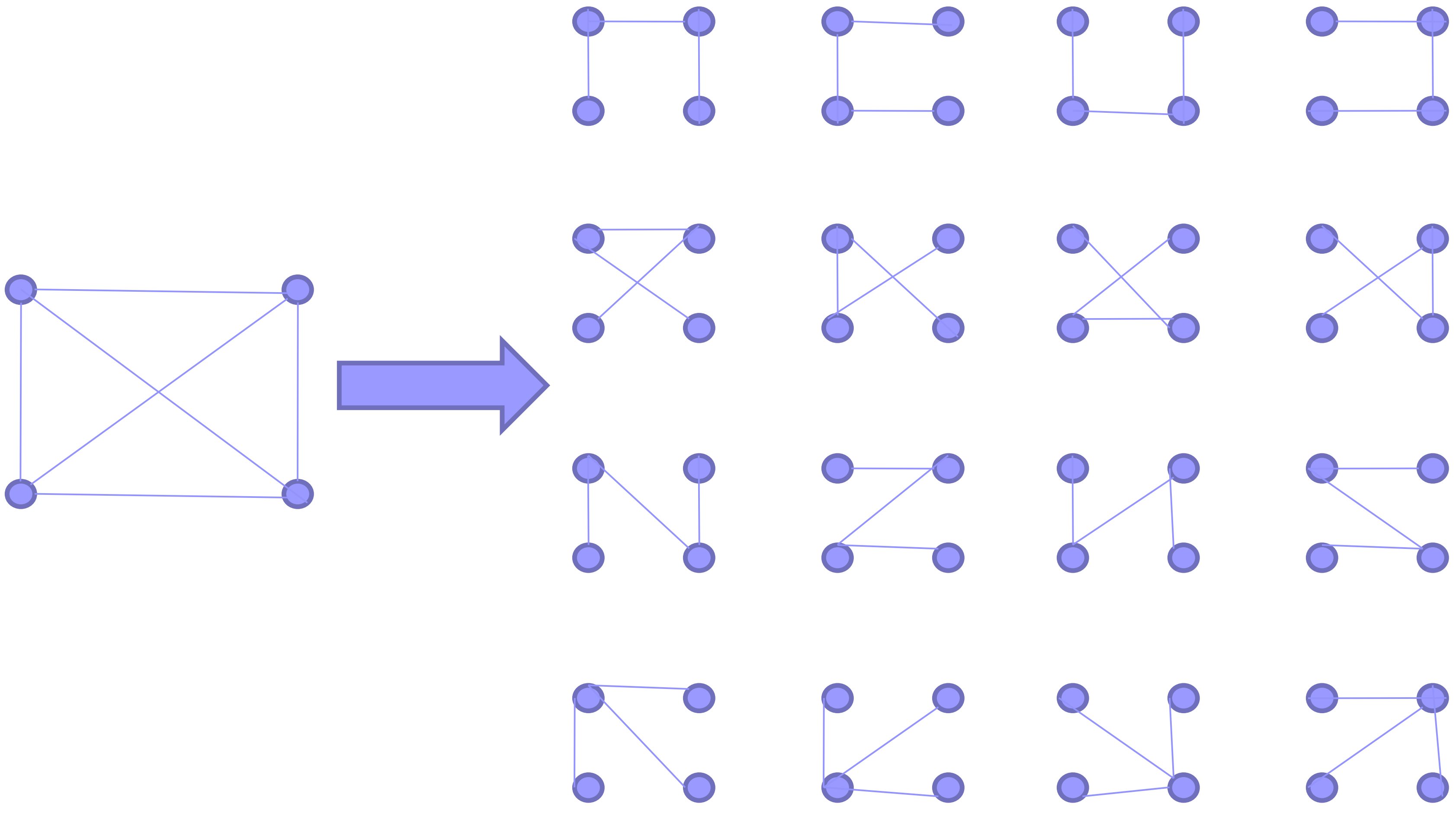
$$(V_1-1)+(V_2-1)+\ldots+(V_k-1) = m$$

相减即可

生成树(Spanning Tree)、生成森林

- 如果树T是无向图G的生成子图，则称T为G的生成树。
- 如果森林F是无向图G的生成子图，则称F为G的生成森林。

生成树(Spanning Tree)示例



找连通图的生成树方法

找连通图 G 的生成树的方法： “避圈法” 与 “破圈法”

1、避圈法：

添加 e_1，…，e_i，在添加的每一步均保证：e_{i+1}不与 $\{e_1, \ldots, e_i\}$ 的任何子集构成回路。

2、破圈法：在 G_0 (即G)，G_1，G_2，… 中去掉 e_1，e_2，e_3，…，其中 e_i 为 G_{i-1} 中某条回路中的边。

即把G中的所有回路均挑破！！！

最小生成树-Minimum Spanning Tree (MST)

设〈G，W〉是加权图，G′⊆ G。

G′中所有边的加权长度之和 称为 G′的加权长度。

设 G 是连通无向图，〈G，W〉是加权图，

G 的所有生成树中加权长度最小者称为〈G，W〉的 最小生成树。

Minimum Spanning Tree (MST)（最小生成树）

贪心法求解最小生成树常用的有两种算法，分别是Prim’s MST algorithm和Kruskal′s MST algorithm(prim算法和kruskal算法). Prim算法是基于点的，而Kruskal算法是基于边的。

最小生成树求法（避圈法、破圈法）

按“避圈法”求最小生成树：设 G 是有m条边的n阶连通无向图，

1° 把 G 的m条边按加权长度递增的顺序排成 e_1，e_2，…，e_m；

2° $T \leftarrow \varnothing$；

3° $j \leftarrow 1$，$i \leftarrow 1$；

（i记录正在扫描的边的下标；j记录T中边数是否已达n-1）

4° 若 $j = n$ 则算法结束。

5° 若 G 的以 $T \cup \{e_i\}$ 为边集合的子图没有回路，

则 $T \leftarrow T \cup \{e_i\}$ 且 $j \leftarrow j+1$；

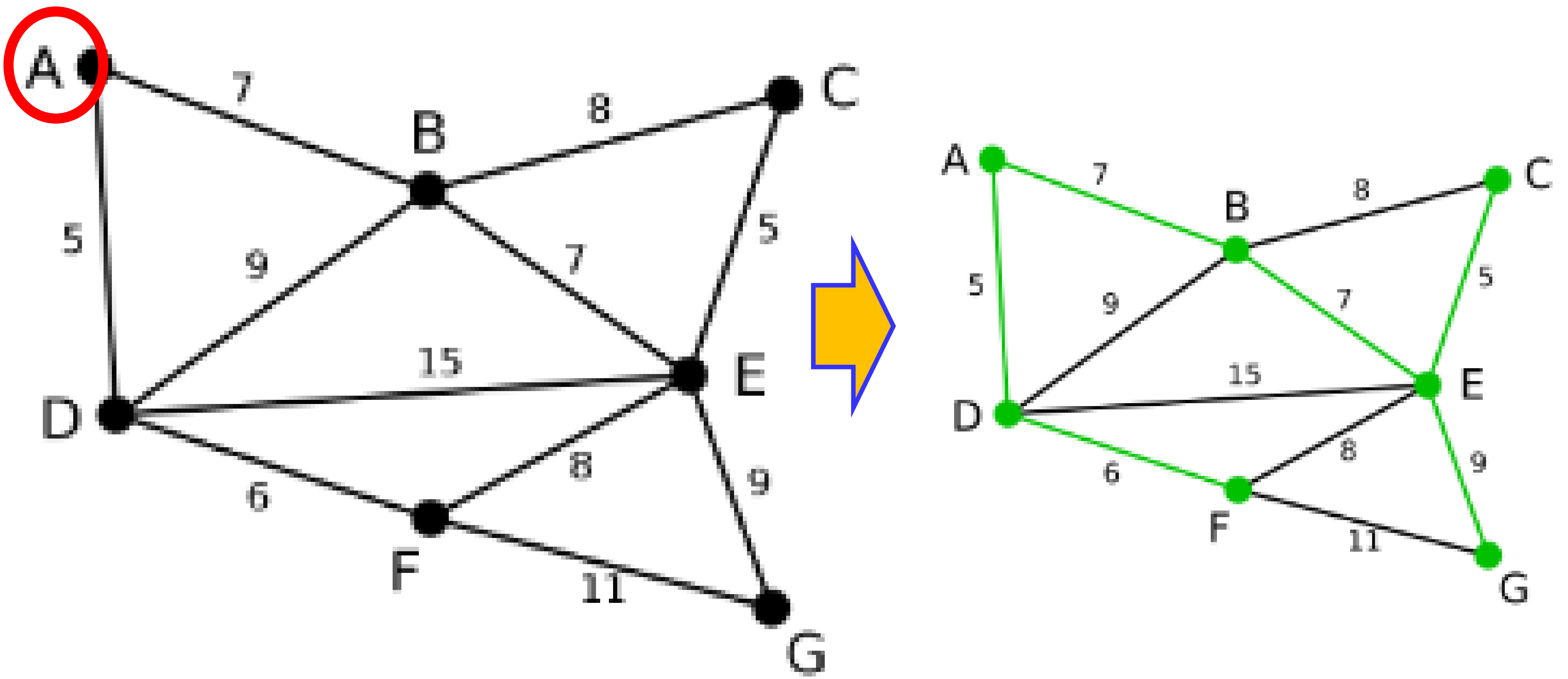
6° $i \leftarrow i+1$，转向 4°

算法结束时，T 即为所求的最小生成树的边集。

最小生树算法--Prim算法

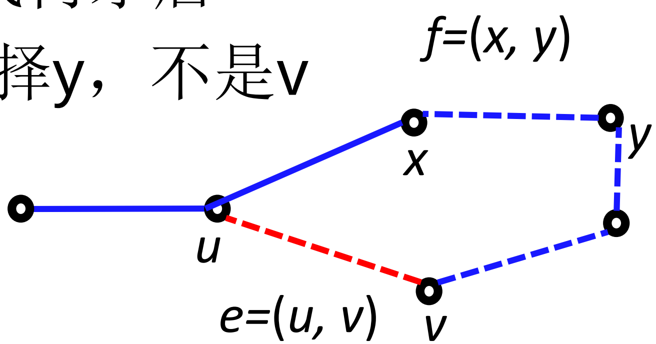
- 用于连通无向图，贪心算法
  - 维护Tree结构
  - 初始E={} V={v} //任取节点v
  - 循环n-1次
    - 选择一条边(v1, v2), 满足
      - $v1 \in V$, $v2 \notin V$,
      - (v1, v2)权值最小
    - $E=E \cup (v1, v2)$
    - $V=V \cup \{v2\}$

Prim算法：示例



Prim算法：正确性证明

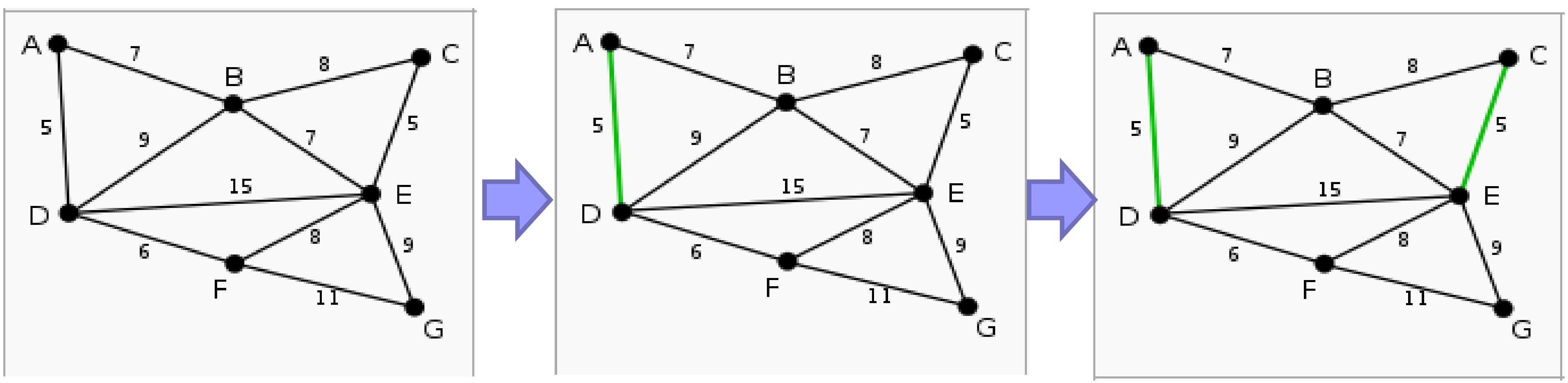
- 假设Prim算法得到树P，最小生成树为T
  - 假设在第K步时，Prim算法生成P’ 选择一条边 *e*=(u, v)不在T中.
  - 则在T中存在u，v的一条路径, 其中*f*=(x, y)中节点分别属于P’和T
  - 分两种情况
    - 当w(e) < f(e) ,则和T是最小生成树矛盾
    - 当w(e) > f(e), Prim算法应该选择y，不是v
    - 等于情况，也可选择f。
  - 所以假设不成立。



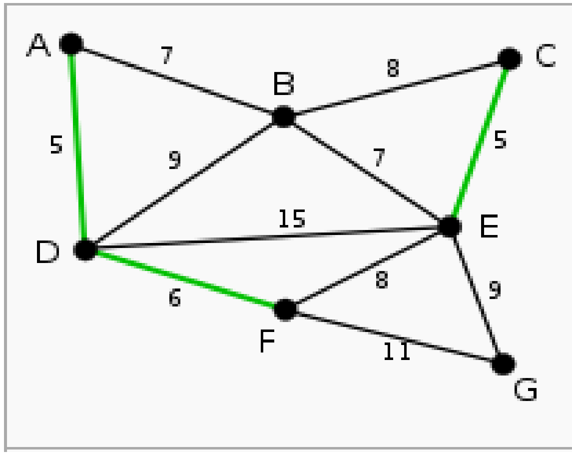
Kruskal算法

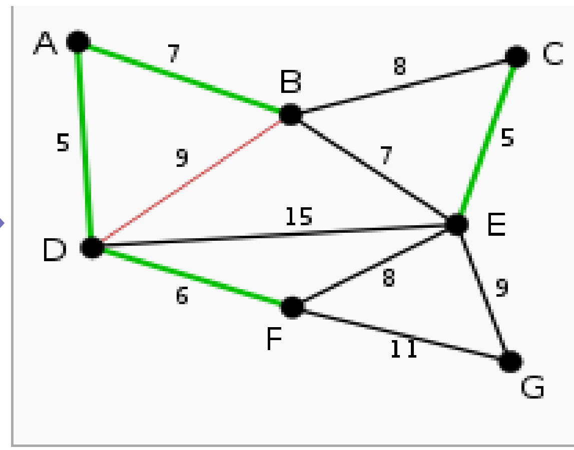
- 贪心算法
  - 将边按从小到大排序
  - 按顺序选择每条边，只要与已选择边不构成圈，就选择。
  - 终止条件
    - 已经选择了n-1条边；
    - 如果处理所有边，仍然不够n-1条，则说明图不连通

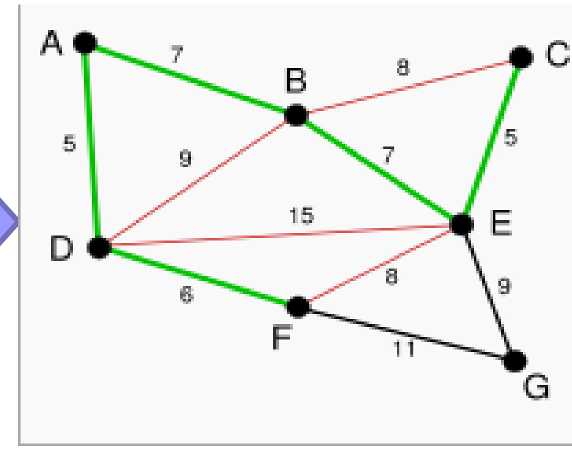
Kruskal's Algorithm - Example



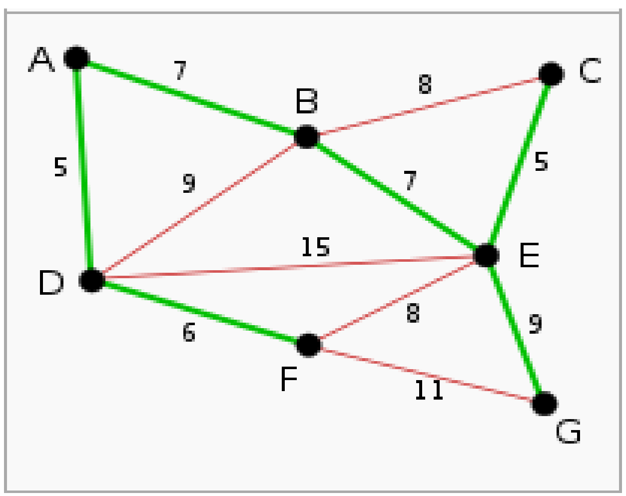
Kruskal's Algorithm - Example







下一步就是关键了。下面选那条边？ BC或EF吗？



枝、弦

设 T 是连通无向图 G 的生成树，称 T 的边为枝，而 G 的不属于 T 的边称为 弦。

连通图 G 的边 e 是枝还是弦？

与给定的生成树 T 密切相关。

- 对于 G 的某个生成树 T， e 是枝，而对于 G 的另一个生成树 T_1， e 却可能是弦。
- 但是，对于 G 的任何生成树，枝的数目和 弦的数目都是固定的。

定理7.6.5

设 G 是有 m 条边的 n 阶连通无向图，则对于 G 的任何生成树 T，都有 $n-1$ 个枝和 $m-n+1$ 个弦。

圈秩、余圈秩

设 n 阶无向图 G 有 m 条边和 k 个分支，

定义 G 的余圈秩 $r = n - k$，圈秩$\mu = m - n + k$。

显然，如果 G 是连通图，则

G 的余圈秩 r 是枝的数目， 圈秩 μ是弦的数目。

基本回路（圈）

由定理7.6.1知，如果在生成树中增加一条弦，

则恰产生一个回路。

定义7.6.7（基本回路）设 T 是连通无向图 G 的生成树，

G 的只包含一条弦的回路称为基本回路。

* 某回路对这个生成树是基本回路，而对另一个生成树却未必是基本回路。

定理7.6.6 生成树

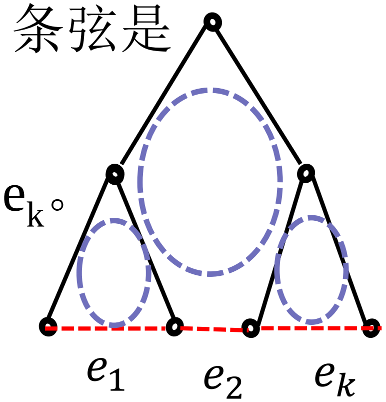
设 T 是连通无向图 G 的任意生成树。

i） 基本回路的数目等于 G 的圈秩μ;

ii) 对于G的任意回路 C，总可以找到若干个基本回路 C_1，C_2，…，C_k，使 C 与 $C_1 \oplus C_2 \oplus \ldots \oplus C_k$ 的差别仅在于孤立点。

证明：i) 显然。

ii) 设 C 是 G 的任意回路，C 包含 k 条弦，显然 k>0，设这 k 条弦是 e_1，e_2，…，e_k，C_i 是包含 e_i 的基本回路 (i =1，2，…，k)。



令 $C' = C_1 \oplus C_2 \oplus \ldots \oplus C_k$，则 C′ 包含的弦也是 e_1，e_2，…，e_k。

因此，$C \oplus C'$ 中的边都是枝，则 $C \oplus C'$ 是非循环的 。

下面证明 $C \oplus C'$ 是零图。

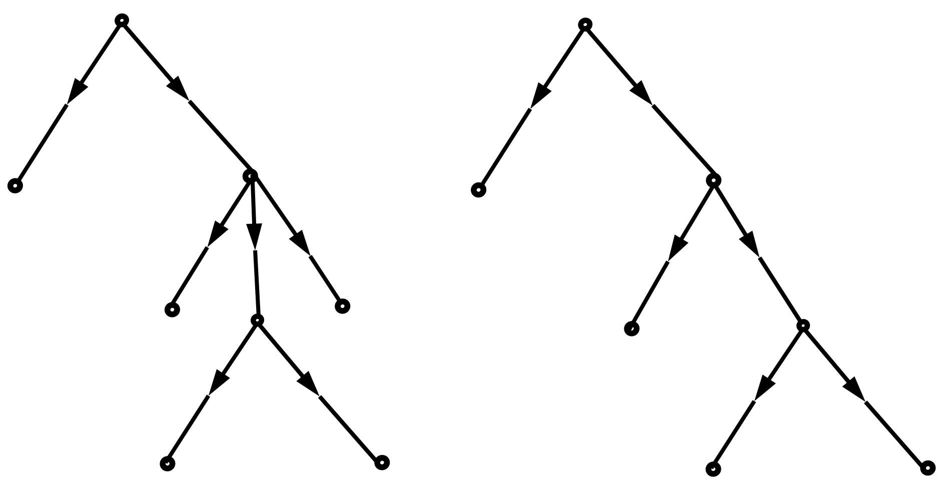
若 $C \oplus C'$ 不是零图，必有一分支是阶大于 1 的树，根据定理7.6.2，$C \oplus C'$ 有端点。

另，因为 C 和 C′ 都是欧拉图，所以$C \oplus C'$ 是欧拉图。这与 $C \oplus C'$ 有端点矛盾，故 $C \oplus C'$ 必为零图，即C 与 C′的差别仅在于孤立点。

有向树

一个结点的入度为 0，其余结点的入度均为 1 的弱连通有向图　称为有向树。

在有向树中，入度为 0 的结点称为根，出度为 0 的结点称为叶，出度大于 0 的结点称为分支结点， 从根至任意结点的距离称为该结点的级， 所有结点的级的最大值称为有向树的高度。



(a)　图7.6.4 有向树 (b)

定理7.6.7及其证明

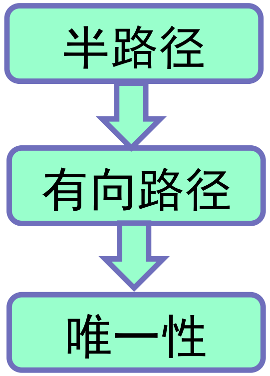
设 v_0 是有向图 D 的结点。　D 是以 v_0 为根的有向树　iff

从 v_0 至 D 的任意结点 恰有一条路径。

证明：⇒）设 D ＝〈V， E，Ψ〉是有向树，v_0 是D的根。

因为 D是弱连通的，任取 $v'\in V$，则存在从 v_0 至 v'的半路径 P，

设 P为 $v_0\, e_1\, v_1 \ldots v_{p-1}\, e_p\, v_p$，其中 $v_p = v'$。

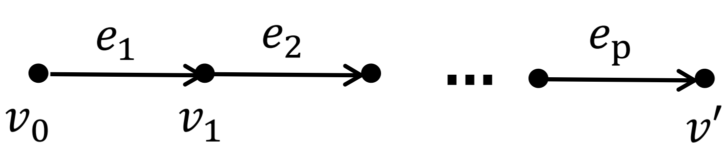


因为 $d_D^-(v_0) = 0$，所以 e_1 是正向边，因为$d_D^-(v_1) =1$，

所以 e_2 也是 正向边。

由归纳法可以证明： 每个 e_i（$1\le i \le p$）均是正向边。

故 P 为有向路径。

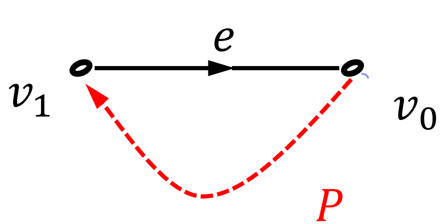


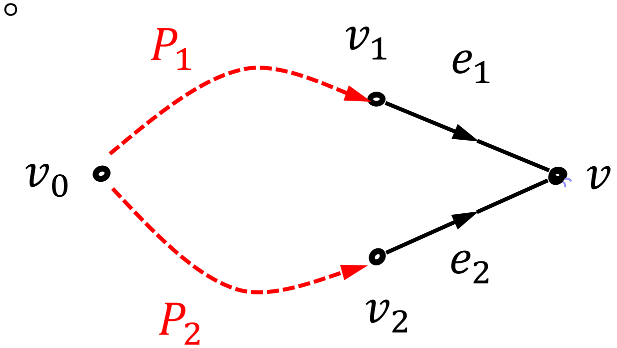
若从v_0至v'有两条路径 P_1 和 P_2，则P_1和P_2至少有一个公共点的入度大于1，与 D 是有向树矛盾。

定理7.6.7证明

证明：$\Leftarrow$）若 $d_D^-(v_0) > 0$，则存在边e以v_0为终点（D弱连通），设v_1是e的起点，P是从v_0至v_1的路径，则在D中存在两条不同的从v_0至v_0的 路径Pv_1ev_0和v_0，与已知条件矛盾，所以 $d_D^-(v_0) = 0$。

若$d_D^-(v) >1$，其中 v 是D的结点，则存在两条边e_1和e_2以v为终点。设e_1和e_2的起点分别是v_1和v_2，从v_0至v_1和从v_0至v_2的路径分别是P_1和P_2，则P_1e_1v和P_2e_2v是两条不同的从v_0至v的路径，与已知条件矛盾。所以，D是有向树，且v_0是D的根。





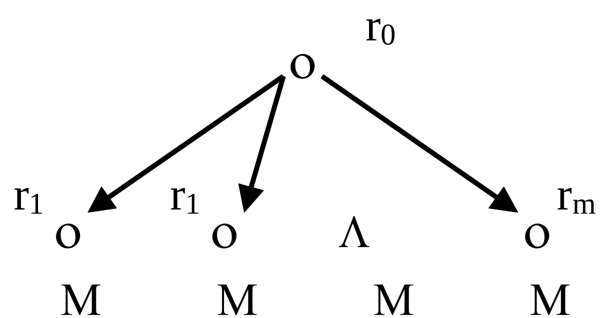
有向树的归纳定义

也可用归纳法定义有向树。

定义7.6.9 有向树归纳定义如下：

i) o

ii)



- $D_1, D_2, \ldots, D_m$分别是以$r_1, r_2, \ldots, r_m$为根的有向树，并且两两不 相交。
- r_0不是$\cup D_i$中节点
- $e_1, e_2, \ldots, e_m$不是$\cup D_i$中的边
- $e_1, e_2, \ldots, e_m$是$<r_0, r_i>$

$D=G \cup (\cup D_i)$是有向树，r0是根，Di是D的子树

有向森林

每个弱分支都是有向树的有向图，

称为有向森林。

m元有向树

设 m∈N，D 为有向树。

i) 如果D的所有结点出度的最大值为m，则称D为m元有向树。

ii) 如果对于m元有向树D的每个结点v，皆有$d_D^+(v) = m$ 或 $d_D^+(v) = 0$，则称D为完全m元有向树。

完全二元有向树也称二叉树。

用途：字母和符号识别程序　　{+，-，*，/}

00　01　10　11

统计字母出现的频繁程度

叶加权二叉树

设 V 是二叉树 D 的叶的集合，R_+ 是全体正实数的集合，$W: V \to R_+$，则称〈D，W〉为叶加权二叉树。

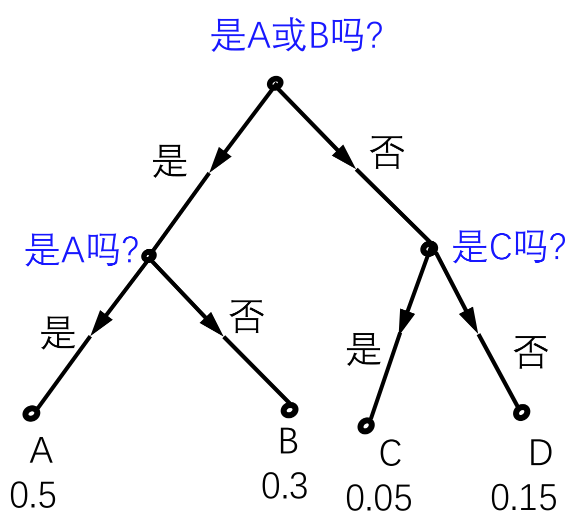
对于 D 的任意叶 v，称 W(v) 为 v 的权，

$\Sigma_{v\in V}$ (W (v) * L(v)) 称为〈D，W〉的叶加权路径长度，其中 L(v) 为 v 的级。

* 我们用叶表示字母或符号，用分支结点表示判断，用权表示字母或符号出现的概率，则叶加权路径长度就表示算法的平均执行时间。

最优二叉树

设〈D，W〉是叶加权二叉树。如果对任一叶加权二叉树 〈D′，W′〉，只要对于任意正实数r，D 和 D′中权等于 r 的叶的数目相同，就有〈D，W〉的叶加权路径长度不大于 〈D′，W′〉的叶加权路径长度，则称〈D，W〉为最优的。



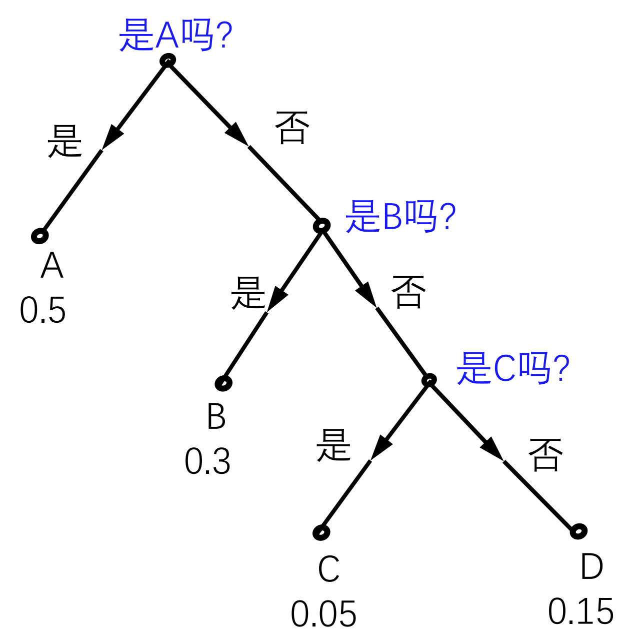


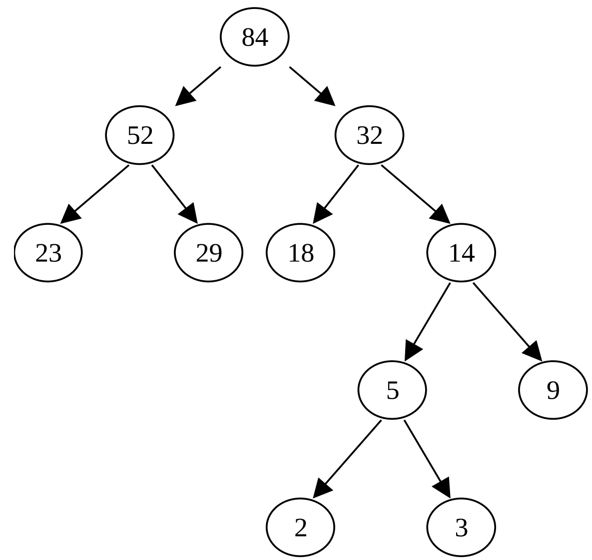
图7.6.5 用叶加权二叉树研究算法

最优二叉树求取

* 我们把求某问题的最佳算法归结为求最优二叉树。

最优二叉树求取算法：（举例说明）

2	3	9	18	23	29
	5	9	18	23	29
		14	18	23	29
			32	23	29
			32		52
					84



- 将求 n 个叶的最优二叉树归结为求 n-1 个叶的最优二叉树。
- 所有分支结点中的数值之和就是叶加权路径长度

有序树、有序森林

为每一级上的结点规定了次序的有向树称为有序树。如果有向森林F的每个弱分支都是有序树，并且也为F的每个弱分支规定了次序，则称F为有序森林。

可以用有序树表示算术表达式，其中 叶表示参加运算的数或变量，分支结点表示运算符

有序树、有序森林

我们约定，在画有序树时，总是把根画在上部，并规定同一级上结点的次序是从左至右。在画有序森林时，弱分支的次序也是从左至右。

例：用有序树表示算术表达式

$v_1 * v_2 + v_3 * (v_4 + v_5 / v_6)$

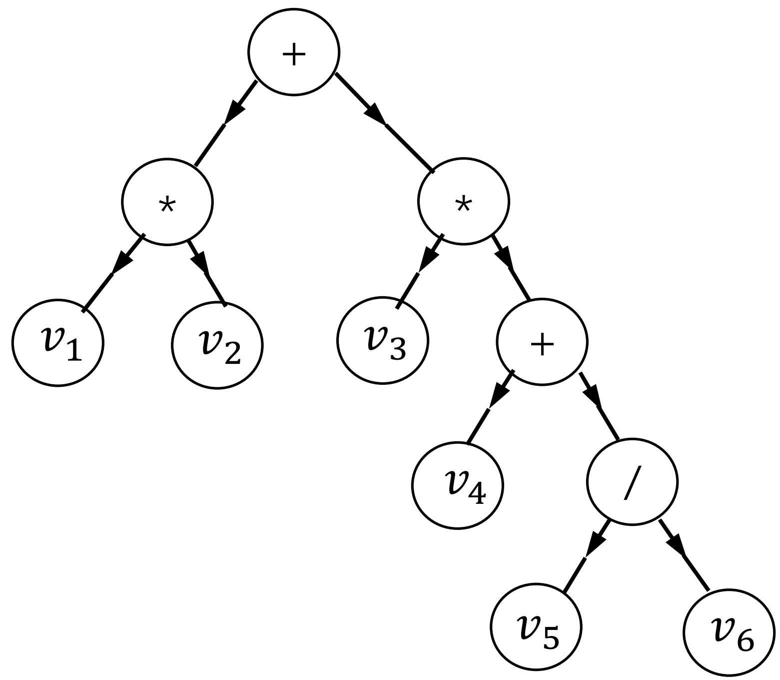
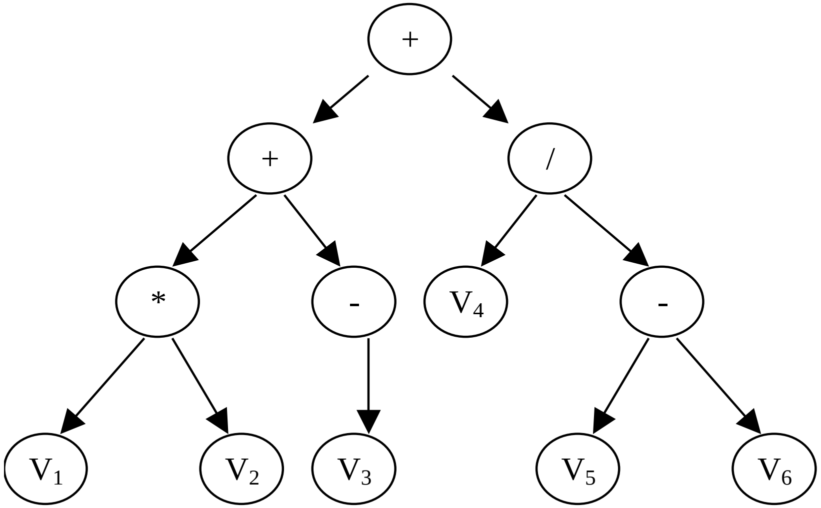


图7.6.8　用有序树表示算术表达式

? 有序树、有序森林(例题)

例：用有序树表示算术表达式

$$((v_1 * v_2) + (-v_3)) + v_4 / (v_5 - v_6)$$



定位有序树

为每个分支结点的儿子规定了位置的有序树称为定位有序树。

定位有序树

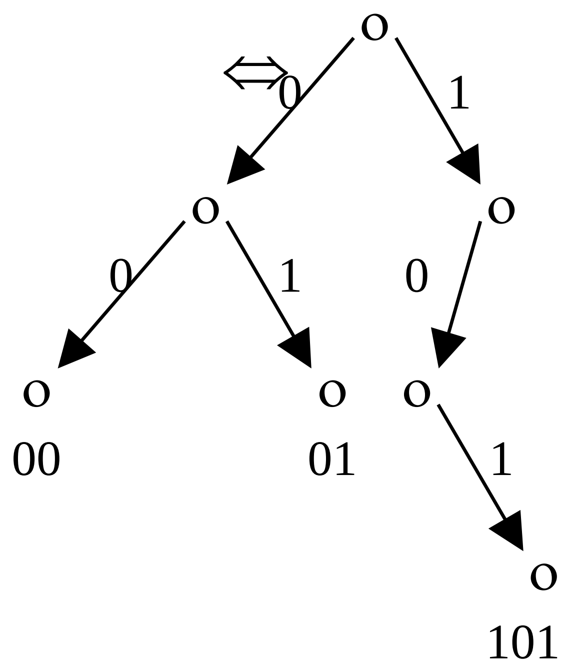
例：可用定位二元有序树 表示 二进制编码情况。

T 中全体叶的编码表示的集合称为 T 的前缀编码。

{00，01，101}

计算机存储

⇔



定位有序树

例：在计算机通信中要传输A, B, C, D, E, F, G, H八个字母，他们出现频率为A:30%, B:20%, C:15%, D:10%, E:10%, F:6%, G:5%, H:4%。给出一个最佳编码，使得通讯中出现的二进制数字尽可能少。

定位有序树

可以用定位二元有序树 表示有序森林。

有序森林和定位二元有序树之间可以建立一一对应关系。

有序森林 F	定位二有序树 T
v_1是v_2的长子	v_1是v_2的左儿子
v_1是v_2的大弟	v_1是v_2的右儿子

亲弟弟，不包括堂兄弟

定位有序树

有序森林和定位二元有序树之间可以建立一一对应关系。

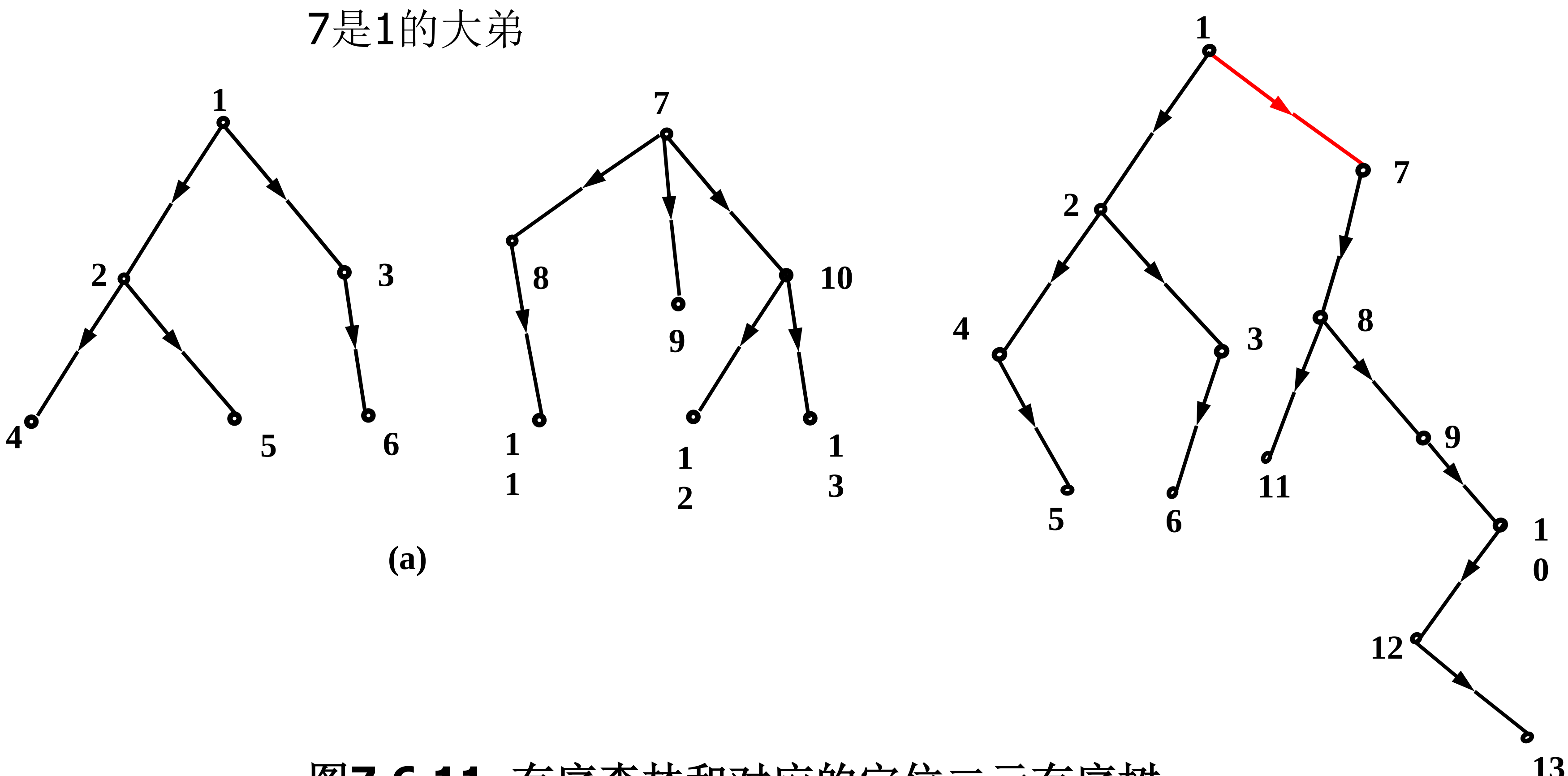


图7.6.11 有序森林和对应的定位二元有序树

思考题

习题7.6 ALL

8、任何二叉树均有奇数个结点。

9、证明 n 阶二叉树有(n+1)/2个叶，其高度 h 满足:

$$\log_2(n+1)-1 \le h \le (n-1)/2$$

6. 将森林转换为对应的二叉树，若在二叉树中，结点u是结点v的父结点的父结点，则在原来的森林中，u和v可能具有的关系是

I. 父子关系　　II. 兄弟关系　　III. u的父结点与v的父结点是兄弟关系

A. 只有II　　B. I和II　　C. I和III　　D. I、II和III