第2篇 图论

第5章 树

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

本章节目录

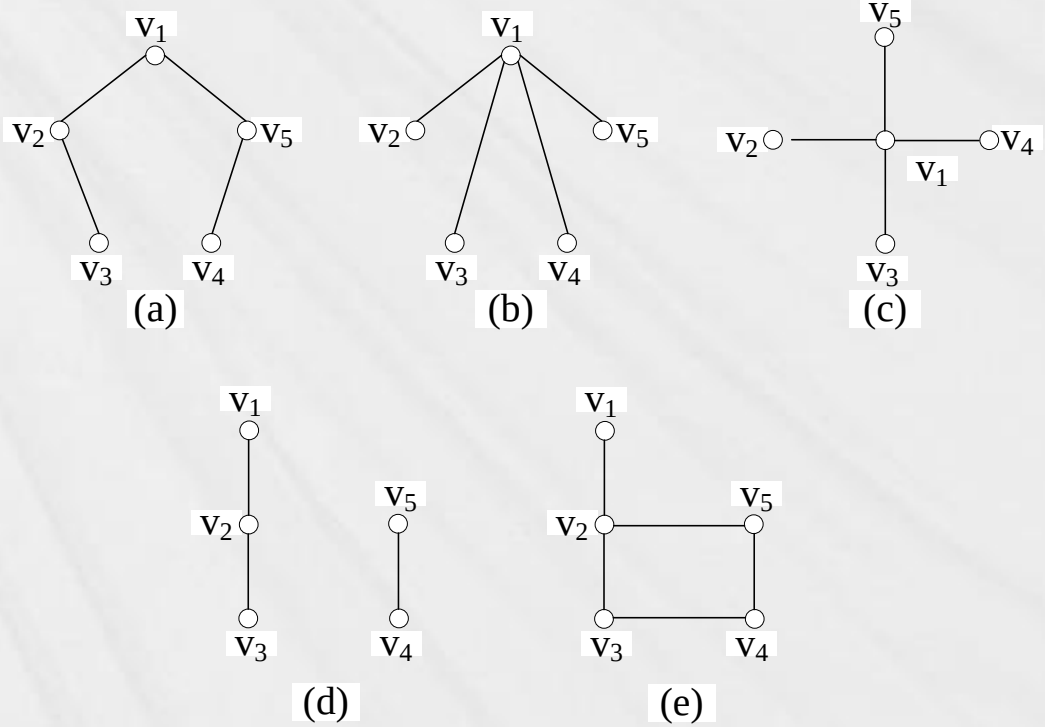
5.1 树与图的生成树

5.1.1树的概念与性质

- 定义5.1 无回路的无向连通图称为无向树，简称为树，记作T。
  - 在树中，度为1的结点称为叶结点；度大于1的结点称为内结点或分支结点。
  - 若图的每个连通分图是树，则称该图为森林。平凡图称为平凡树。

5.1 树与图的生成树

- 下图中，（a）、（b）、（c）连通且无回路，因而是树，并且，（b）和（c）同构；（d）不连通，（e）有回路，因而它们都不是树。在树（a）中，v3、v4是叶结点，v1、v2、v5是分支结点；图（d）是由两棵树构成的森林。



5.1 树与图的生成树

- 定理5.1 设图T =（V，E）是树，则|E| = |V| – 1。

- 定理5.2 图G =（V，E）是树的充分必要条件是图G的每对结点间有且仅有一条路径。
  - 推论 去掉树中任何一条边，树就成了森林；在树中添加任何一条边，就会产生一条回路。

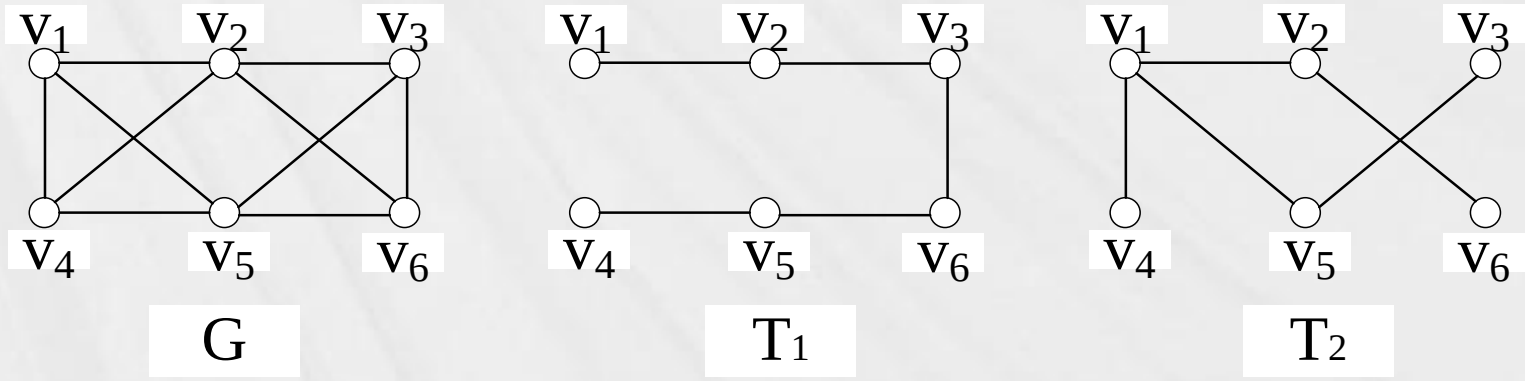
- 定理5.3 任何一棵非平凡树中至少有两个叶结点。

5.1 树与图的生成树

5.1.2图的生成树

1.生成树

- 定义5.2 若无向图G =（V，E）的生成子图T是树，则称T为G的生成树。
  - 下图中，T1、T2都是图G的生成树。



5.1 树与图的生成树

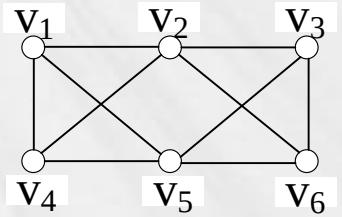
- 定理5.4 无向简单图G =（V，E）存在生成树的充分必要条件是G是连通的。
  - 推论设G = (V，E)是连通的（n，m）图，则m≥n–1。

- 求连通图生成树的一种方法——破圈法：每次从连通图G的回路中删去一条边，直至得到生成树为止，对于（n，m）图，共需删去m – n + 1条边。

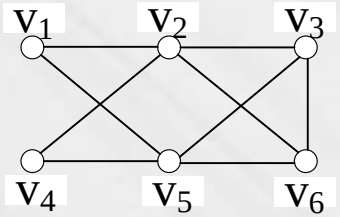
- 求连通图生成树的另一个方法——避圈法，即，每次选择图中一条不会与已选择的边构成回路的边，直至得到生成树为止。对于（n，m）图，共需选择n – 1条边。

5.1 树与图的生成树

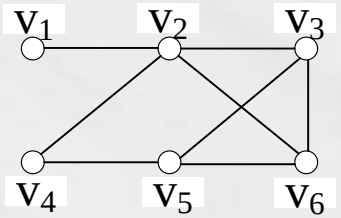
- 对下面的图(a)，由于共有6个结点10条边，采用破圈法得到该图的生成树需要删去5条边，构造生成树的过程如图（b）~（f）所示。



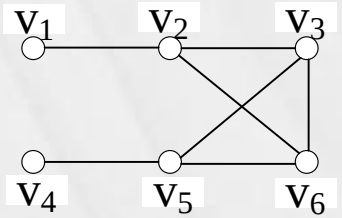
(a)



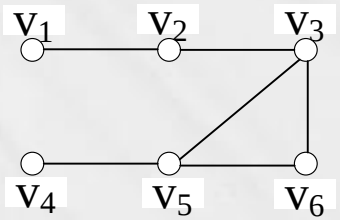
(b)



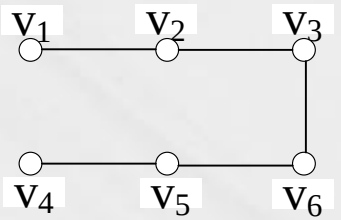
(c)



(d)



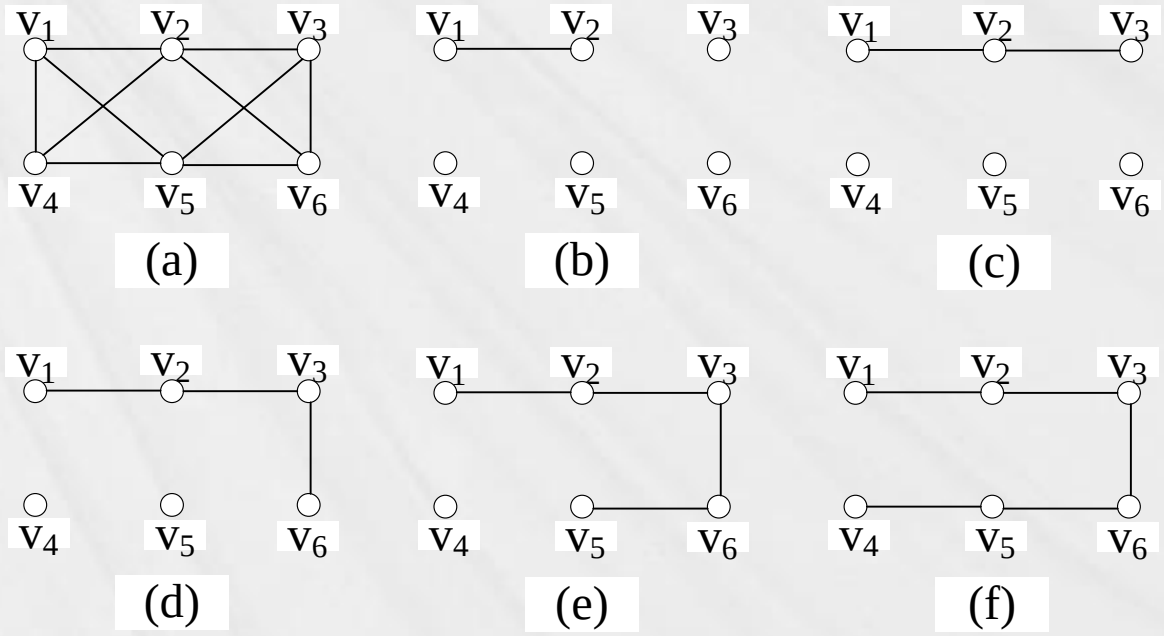
(e)



(f)

5.1 树与图的生成树

- 对下面的图(a)，采用避圈法得到该图的生成树需要选择5条边，构造生成树的过程如图（b）~（f）所示。



5.1 树与图的生成树

2.最小生成树

- 定义5.3 设G =（V，E，W）是加权连通图，T是G的一棵生成树，T的各边的权之和称为T的权，记作　w（T），即，w（T）= $\sum_{e\in T} w(e)$。G的所有生成树中权最小的生成树称为最小生成树。

- 最小生成树的性质：设G =（V，E，W）是一个连通图，T =（V’，E’，W’）是正在构造的最小生成树，若边（u，v）是G中所有一端在V’中、另一端在V – V’中权值最小的一条边，则存在一棵包含边（u，v）的最小生成树。

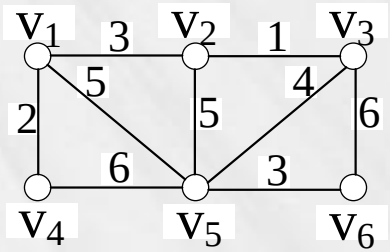
5.1 树与图的生成树

- 普里姆算法：通过一系列扩展子树的过程构造最小生成树。最初的子树只包含图G=（V，E，W）中某一结点，扩展时不断选择连接树中结点与非树中结点的权值最小的边，将该边及结点加入树中，直至得到最小生成树T=（V’，E’，W’）。

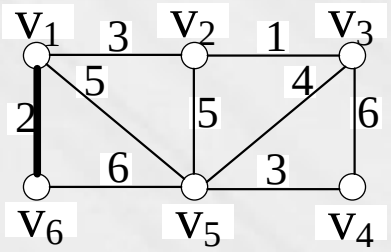
- 克鲁斯卡尔算法：通过一系列扩展子图的过程构造最小生成树。最初的子图是由图G =（V，E，W）中所有结点构成的零图，扩展时选择图中权最小且添加到子图中不会产生回路的一条边，将该边加入子图中，直至得到最小生成树T=（V’，E’，W’）。克鲁斯卡尔算法实际上是一种避圈法。

5.1 树与图的生成树

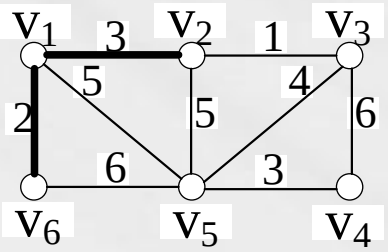
- 对下面的图(a)，采用普里姆算法以v1为最初子树的结点，构造最小生成树的过程如下图（b）~（f）中粗实线所示，w（T）= 13。



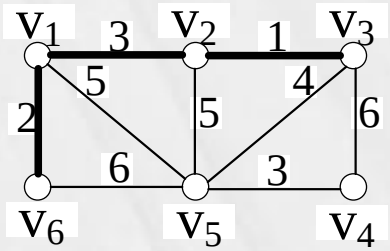
(a)



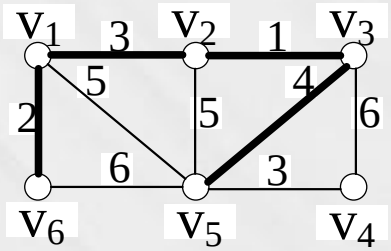
(b)



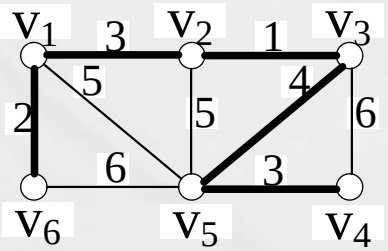
(c)



(d)



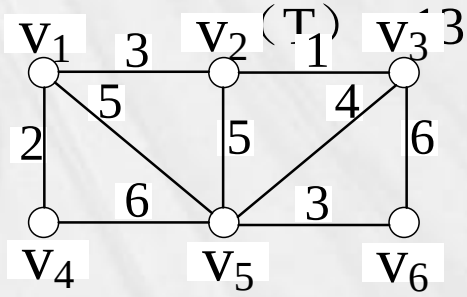
(e)



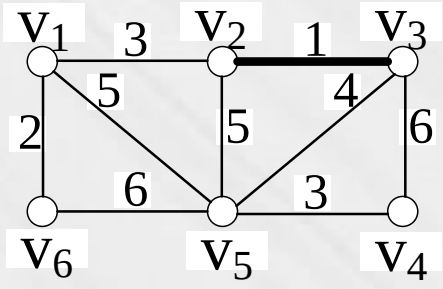
(f)

5.1 树与图的生成树

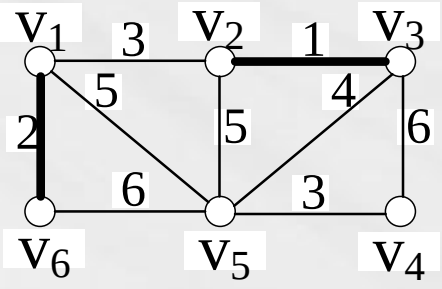
- 对下面的图(a)，采用克鲁斯卡尔算法构造最小生成树的过程如下图（b）~（f）中粗实线所示，w(T)=13。



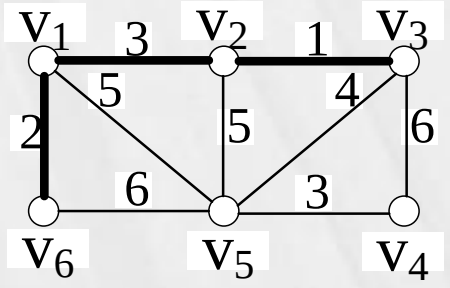
(a)



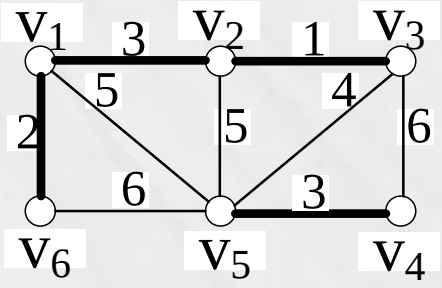
(b)



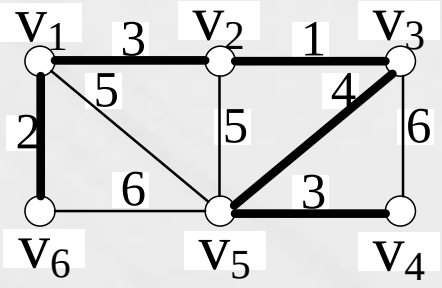
(c)



(d)



(e)

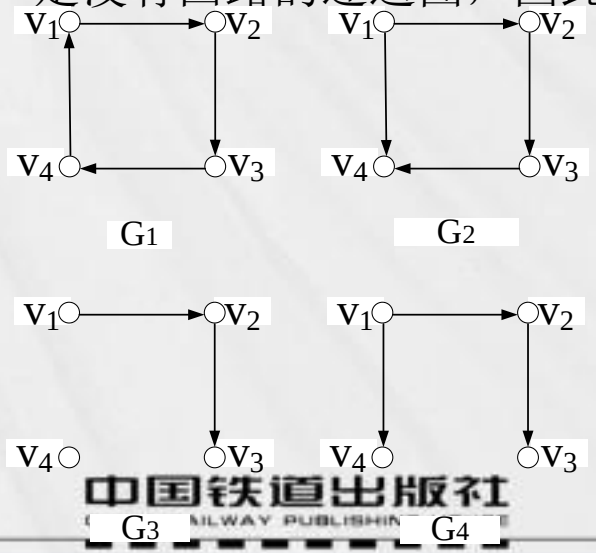


(f)

5.2 根树

5.2.1根树的基本概念

- 定义5.4 如果有向图在不考虑边的方向时是树，则称其为有向树。
  - 对于下面的图，G1和G2在不考虑边的方向时，是一模一样的有回路的连通图，因此，它们都不是有向树。G3不连通，因此也不是有向树。G4在不考虑边的方向时，是没有回路的连通图，因此是有向树。



5.2 根树

- 定义5.5 有向树T如果满足以下条件：

  （1）只有一个结点的入度为0，

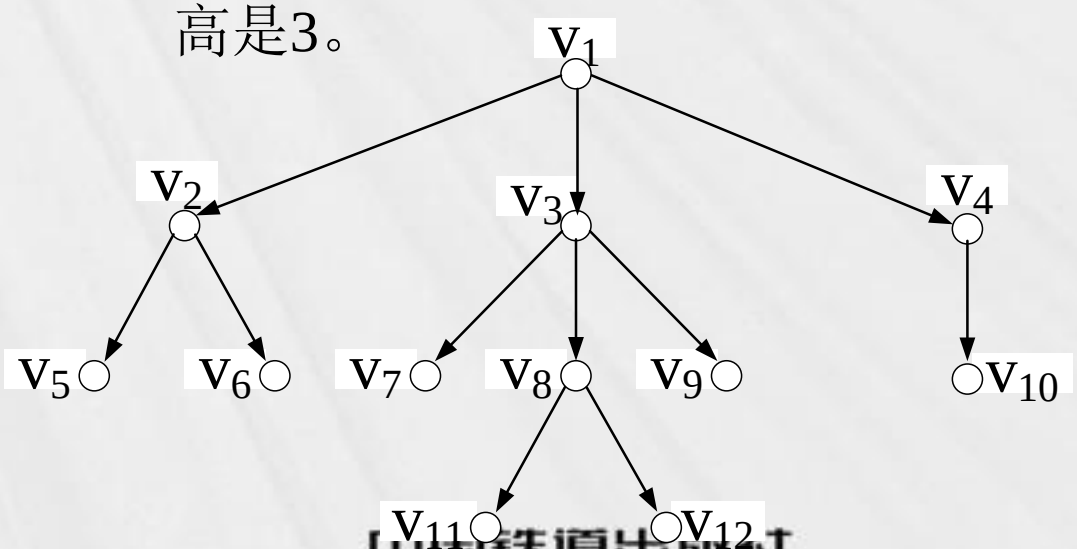
  （2）其它结点的入度都等于1。

则称T为根树，其中，入度为0的结点称为根结点，出度为0的结点称为叶结点，非叶的结点称为内结点或分支结点。

- 上页中的图G4，结点v1是入度为0，其它结点入度都为1，因此它是一棵根树，其中，v1是根结点，v3、v4是叶结点，v2是分支结点。

5.2 根树

- 定义5.6 在根树中，由根到某结点v的路径的长度称为结点v的层数（或级）。所有结点中最大的层数称为根树的高。
  - 下图中的根树，v1的层数是0，v2、v3、v4的层数是1，v5、v6、v7、v8、v9、v10的层数是2，v11、v12的层数是3，该根树的高是3。



5.2 根树

- 根树T =（V，E）有如下的概念：

（1）如果从结点u到v有一条弧，即，<u，v>∈E，则称u为v的双亲，v是u的孩子。

（2）如果u、v同为w的孩子，则称u、v为兄弟。

（3）如果从结点u可达结点v，则称u是v的祖先，v是u的子孙。

（4）由某个结点v及其所有子孙构成的导出子图T’称为T的以v为根的子树。

5.2 根树

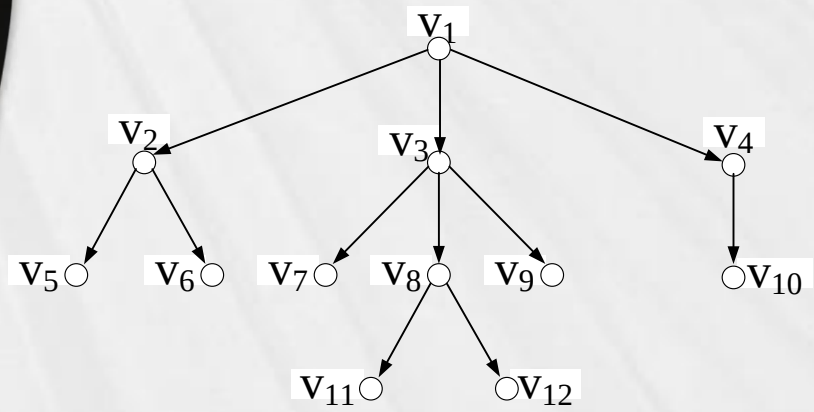
- 对上图中的根树，v1是v2、v3、v4的双亲，v2、v3、v4都是v1的孩子，它们互为兄弟；同理，v2有v5、v6两个孩子，它们互为兄弟；v3有v7、v8、v9三个孩子，它们互为兄弟；而v4只有v10一个孩子，等等。

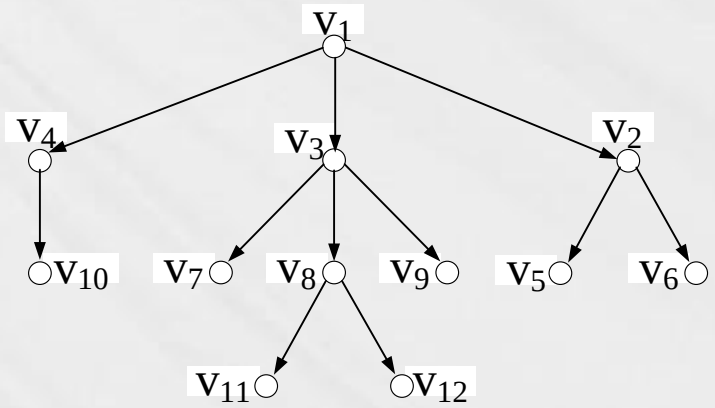
- 从根结点v1可达所有结点，因此，根结点v1是所有结点的祖先，而所有结点都是根结点v1的子孙；并且，每个结点都可达自身，因此，都是自身的祖先和子孙；v3是v3、v7、v8、v9、v11、v12的祖先，而它们是v3的子孙。

- 以v2、v3、v4为根结点，可以得到树T的三棵子树。

5.2 根树

- 定义5.7 在根树中若给定了同一层结点的顺序，则称这样的根树为有序树。
  - 下面两图表示的是同一棵根树，但是两棵不同的有序树。





5.2 根树

5.2.2二叉树

1.m叉树及其性质

- 定义5.8 在根树T中，如果每个结点的出度至多为m，则称T为m叉树；如果每个结点的出度或者为0或者为m，则称T为完全m叉树；如果完全m叉树的所有叶结点都在同一层，则称为满m叉树。

- 定理5.5 设T是一棵完全m叉树，并有n0个叶结点，t个分支结点，则（m – 1）t = n0 – 1。

5.2 根树

2.二叉树及其性质

- 在二叉树中，每个结点至多有两个孩子，分别称为左孩子和右孩子，以左孩子和右孩子分别作为根结点的两棵子树通常称为该结点的左子树和右子树。

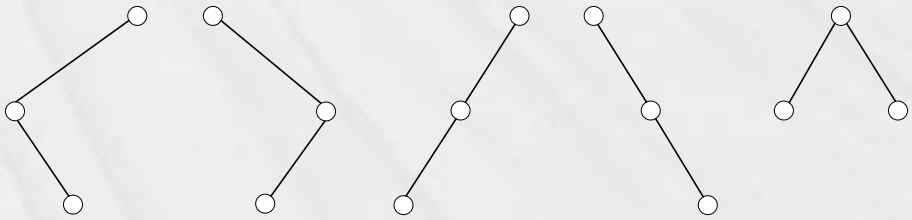
- 定理5.6 设T是一棵二叉树，n0表示叶结点数，n2表示出度为2的结点数，则 n2 = n0 – 1。

5.2 根树

3.二叉树的同构*

- 树是一种特殊的图，因此，树的同构的定义基于图的同构的定义。而根树的同构，除了要满足树同构的条件外，还要保证根结点的对应关系。而二叉树的同构，除了要满足根树同构的条件外，还要保证左右孩子的对应关系。

- 下图中的五棵树作为一般的树而言，它们是同构的；但是作为根树，仅有T1和T2、T3和T4同构，它们均不与T5同构；如果作为二叉树，它们彼此不同构。事实上，由3个结点可以构成的不同构的二叉树只有这五种形式。



5.2 根树

5.2.3二叉树的遍历

- 二叉树的先根遍历算法如下：

（1）访问根结点；

（2）对根结点的左子树进行先根遍历；

（3）对根结点的右子树进行先根遍历。

5.2 根树

- 二叉树的中根遍历算法如下：

  （1）对根结点的左子树进行中根遍历；

  （2）访问根结点；

  （3）对根结点的右子树进行中根遍历。

5.2 根树

- 二叉树的后根遍历算法如下：

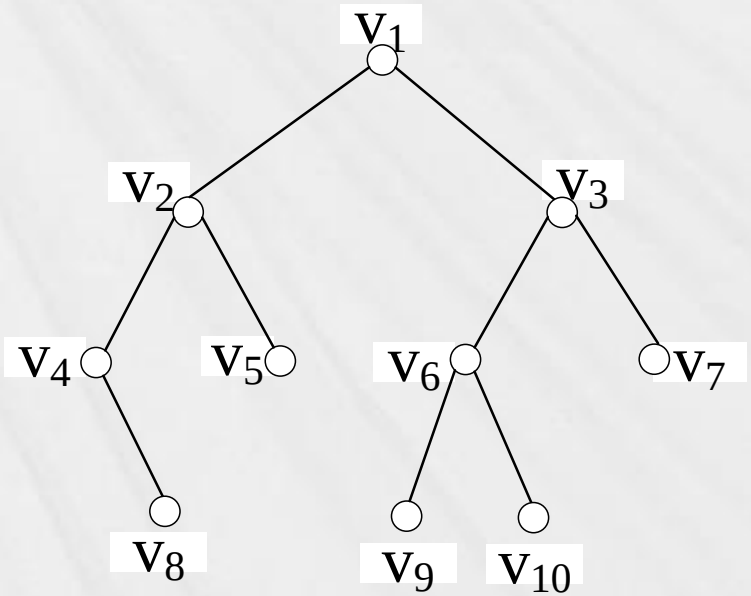
  （1）对根结点的左子树进行后根遍历；

  （2）对根结点的右子树进行后根遍历；

  （3）访问根结点。

5.2 根树

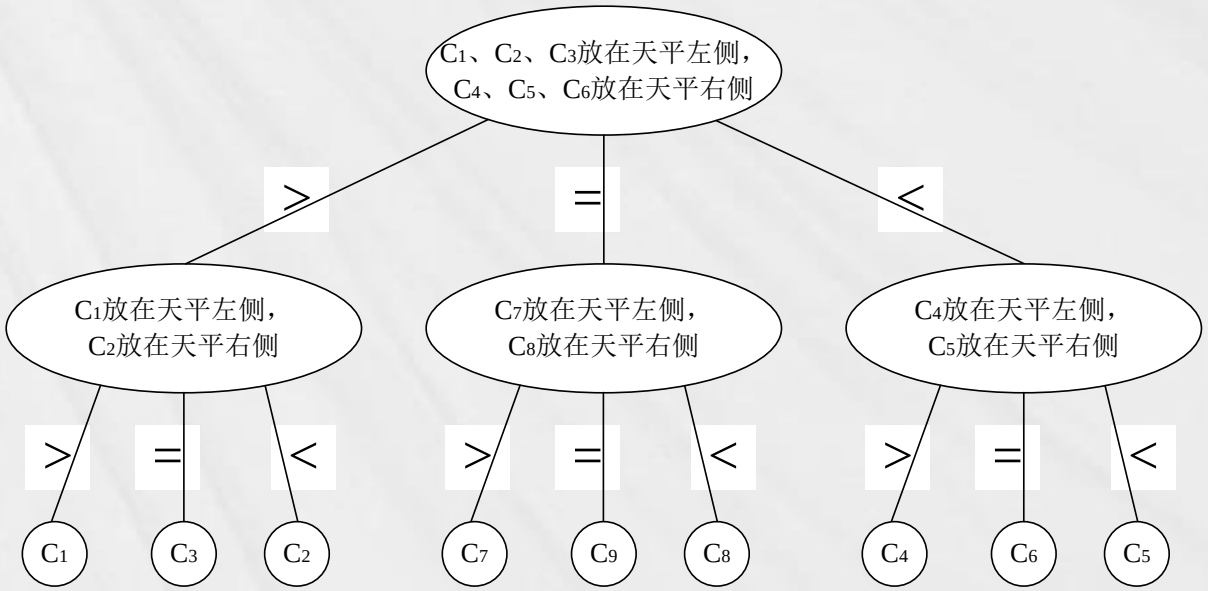
- 对下图中的二叉树，先根遍历访问各结点的顺序依次为：v1 v2 v4 v8 v5 v3 v6 v9 v10 v7；中根遍历访问各结点的顺序依次为：v4 v8 v2 v5 v1 v9 v6 v10 v3 v7；后根遍历访问各结点的顺序依次为：v8 v4 v5 v2 v9 v10 v6 v7 v3 v1。



5.3 树的应用*

5.3.1决策树

- 如果树中每个分支结点会问一个问题，每条边代表对问题的不同回答，从根结点开始，每回答一个问题沿相应的边向下移动，最后到达某个叶结点，即获得一个结果，这样的一棵树称为决策树。



5.3 树的应用*

5.3.2二叉搜索树

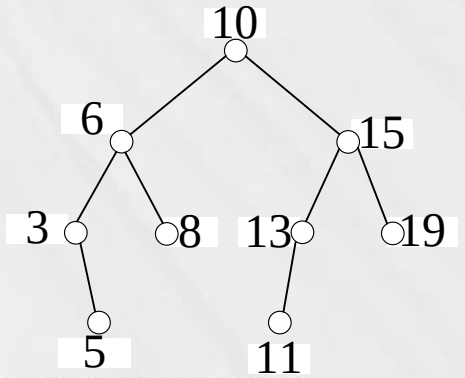
- 定义5.9 二叉搜索树或者是空树；或者是具有下列性质的二叉树：

  （1）如果它的左子树不空，则左子树上各结点的值均小于它的根结点的值；

  （2）如果它的右子树不空，则右子树上各结点的值均大于它的根结点的值；

  （3）它的左、右子树也分别是二叉搜索树。

  - 由数据10，15，6，8，3，13， 5，19，11构成的二叉搜索树如下。



5.3 树的应用*

5.3.3最优二叉树与哈夫曼编码

1.最优二叉树

- 定义5.10 对于二叉树的每个叶结点v都对应一个权w（v），则该二叉树称为加权二叉树。

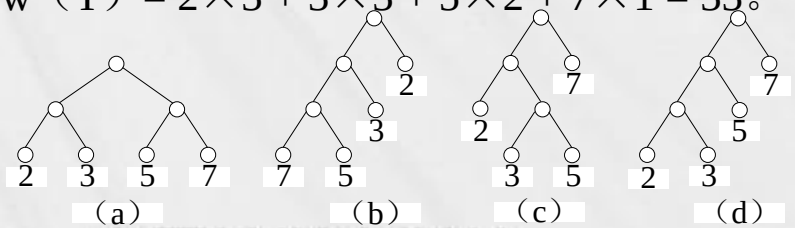
- 定义5.11 在n个叶结点的加权二叉树T中，如果叶结点v_i的层数为 L（v_i），则$\sum_{i=1}^{n} w(v_i)L(v_i)$称为加权二叉树T的权，记作w（T）。
  - 对于下图中的各加权二叉树

（a）w（T）= 2×2 + 3×2 + 5×2 + 7×2 = 34;

（b）w（T）= 7×3 + 5×3 + 3×2 + 2×1 = 44;

（c）w（T）= 2×2 + 3×3 + 5×3 + 7×1 = 35;

（d）w（T）= 2×3 + 3×3 + 5×2 + 7×1 = 33。



5.3 树的应用*

- 定义5.12 在所有叶结点权相同的加权二叉树中，权最小的二叉树称为最优二叉树。

- 若已知n个权值分别为w_1、w_2、…、w_n的结点，以这n个结点为叶结点构造最优二叉树的哈夫曼算法如下：

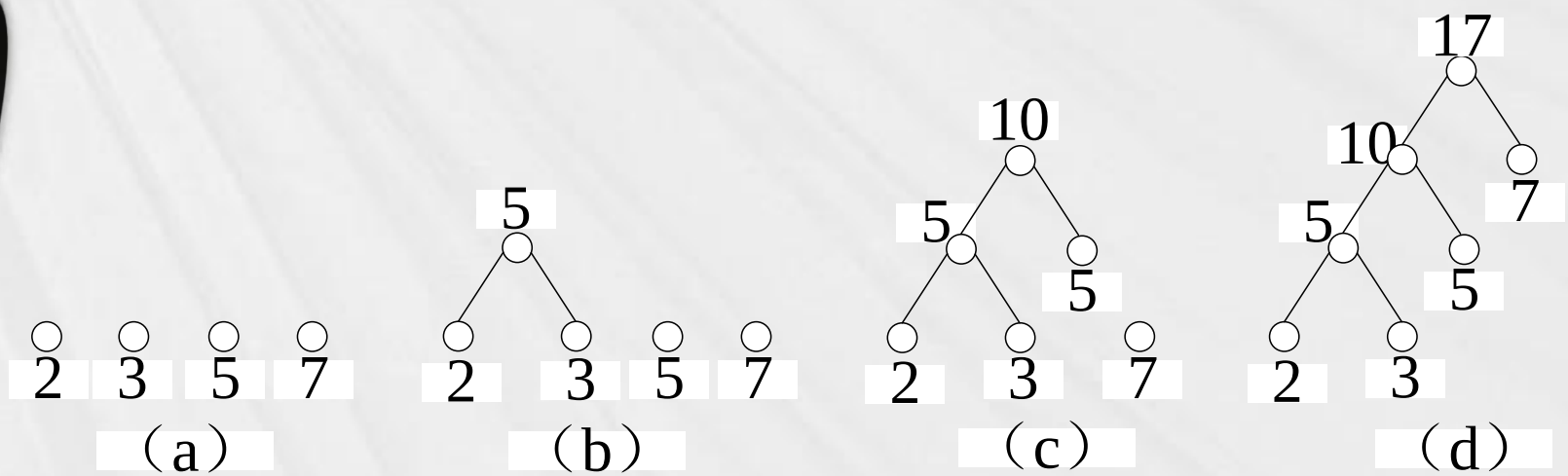
  （1）把这n个叶结点看做n棵仅有根结点的加权二叉树组成的森林。

  （2）在森林中选择权值最小的两棵加权二叉树，以它们作为左右子树构造一棵新的加权二叉树，新树根结点的权值为左右子树根结点权值之和，从森林中删除选择出的这两棵子树，同时把新加权二叉树加入森林。

  （3）重复第（2）步直至森林中只有一棵加权二叉树为止。

5.3 树的应用*

- 对于权值分别为2，3，5，7的四个结点构造最优二叉树具体过程如下图（a）~（d）所示。



5.3 树的应用*

2.哈夫曼编码

- 定义5.13 设$a_1a_2\ldots a_n$是长度为n的字符串，称其子串a_1，a_1a_2，…，$a_1a_2\ldots a_{n-1}$是$a_1a_2\ldots a_n$的长度分别为1，2，…，$n-1$的前缀。

- 定义5.14 设$A=\{b_1, b_2, \ldots, b_n\}$是一个字符串的集合，若对于任意的$b_i$，$b_j \in A$，$b_i \neq b_j$，$b_i$和$b_j$不互为前缀，则称A为一个前缀码。

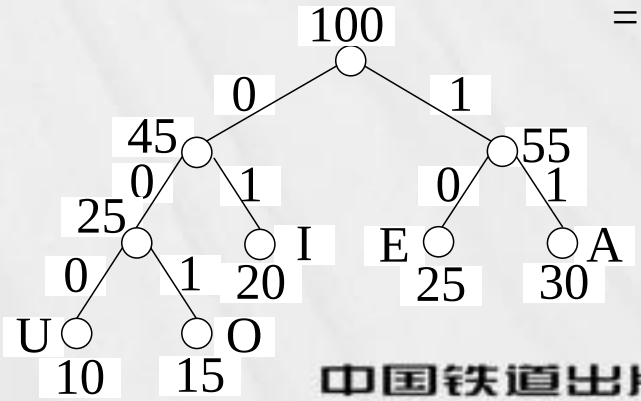
- 最优二叉树可用于构造使传送字符串的编码长度最短的前缀码，具体方法如下：

  - 设需要编码的字符集为$\{c_1, c_2, \ldots, c_n\}$，各个字符相应的使用频率为$w_1$，$w_2$，…，$w_n$，以$c_1$，$c_2$，…，$c_n$为叶结点，以$w_1$，$w_2$，…，$w_n$为叶结点的权构造最优二叉树，规定在最优二叉树中每个结点到其左孩子的边上标0，到其右孩子的边上标1，则从根结点到每个叶结点所经过的边上相应的0和1组成的序列就是该结点对应字符的编码。这样的编码我们称为哈夫曼编码。

5.3 树的应用*

- 假设通讯中，A、E、I、O、U出现的频率分别为30%、25%、20%、15%、 10%。求传输它们的哈夫曼编码。用哈夫曼编码传输1000个按上述频率出现的字符需要多少个二进制位？
  - 带权30，25，20，15，10的最优二叉树如下图所示，为各边标上0、1。
  - 得A、E、I、O、U的哈夫曼编码分别为11、10、01、001、000。
  - 因此，传输1000个这样的字符所用的二进制位为

$$1000\times(30\%\times2+25\%\times2+20\%\times2+15\%\times3+10\%\times3) = 2250$$



本章小结

- 树的概念
  - 树、森林与图的关系
  - 图的生成树
    - 破圈法
    - 避圈法
  - 加权图的最小生成树
    - 普里姆算法
    - 克鲁斯卡尔算法
- 树的相关概念
  - 有向树
  - 根树
  - 有序树
  - 二叉树
- 树的典型应用
  - 决策树
  - 二叉搜索树
  - 最优二叉树与哈夫曼编码