第7章 树

7.1 无向树及生成树

7.2 根树及其应用

7.1 无向树及生成树

一. 无向树的定义

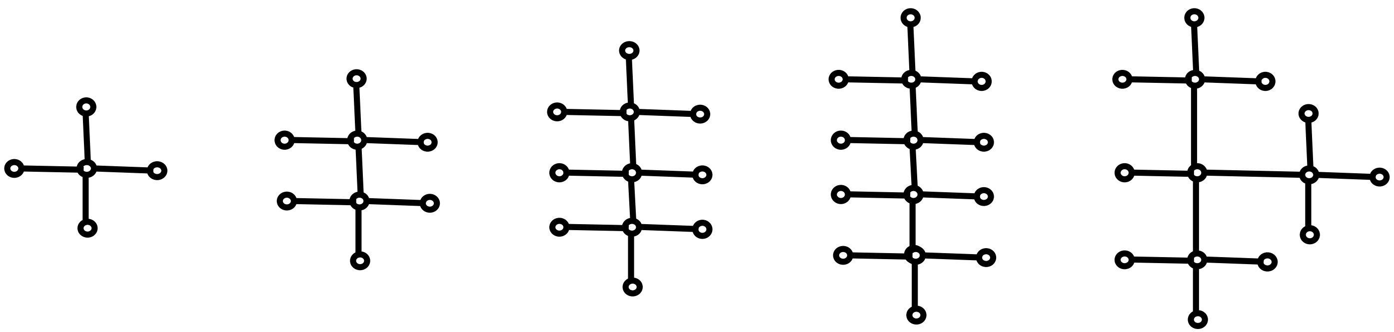
二. 无向树的性质

三. 生成树

四. 最小生成树与避圈法

树的起源

英国数学家凯莱(Arthur Cayley)于19世纪中叶研究饱和碳氢化合物C_nH_{2n+2}的同分异构体时提出树的概念. 当n=1,2,3时, 都只有一棵非同构的树; 当n=4时, 有2棵不同构的树.



甲烷　　乙烷　　丙烷　　丁烷　　异丁烷

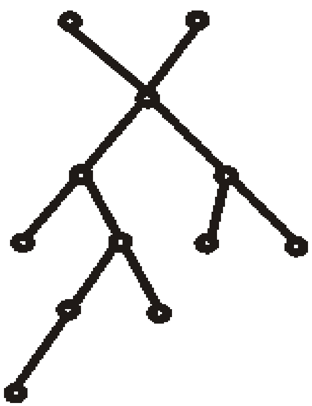
7.1 无向树及生成树

一. 无向树的定义

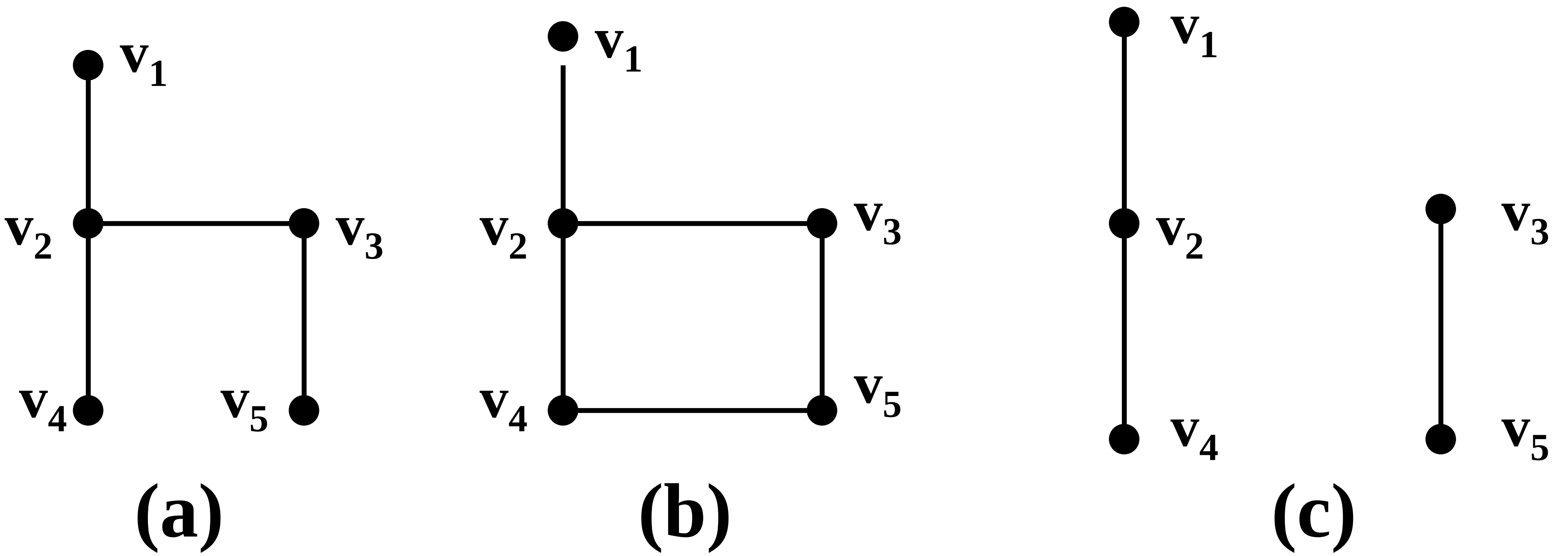
连通无回路的无向图称为无向树，简称树，常用T表示树。(即树是不包含回路的连通图)

- 平凡图称为平凡树。
- 若无向图G至少有两个连通分支，且每个连通分支都是树，则称G为森林。
- 在无向树中，度数为1的顶点称为树叶，度数大于或等于2的顶点称为分支点。

右图为一棵12阶树.



例 7.1 判断下列哪些图是树？



解： 图(a)是树, 因为它连通又不包含回路。

图(b)不是树, 因为图(b)虽连通但有回路,

图(c)不是树，虽无回路但不连通。

在图(a)中, v_1、v_4、v_5为均为树叶,

v_2、v_3均为分支点。

二.无向树的性质

定理 7.1 设$G=<V,E>$,则下面各命题是等价的：

（1）G连通而不含回路（G是树）

（2）G中每对顶点之间存在唯一的路径

（3）G中无回路且 $n = m+1$

（4）G是连通的且 $n = m+1$

(5) G中没有回路，但在任何两个不同的顶点之间加一条新边，在所得的图中得到唯一的一条初级回路。

(6) G是连通的且G中任何边均为桥。

其中n为G中顶点数，m为边数。

定理7.2 设T是n阶非平凡的无向树，则T中**至少有两片树叶**。

证明: 因为T是非平凡树,所以T中每个顶点的度数都大于等于1.

设T有x片树叶, 由握手定理及定理7.1,

$$2(n-1) \geq x + 2(n-x)$$

解得 $x \geq 2$.

◆ 以上两个定理给出了无向树的主要性质，利用这些**性质**和**握手定理**，可以画出阶数n比较小的所有非同构的无向树。

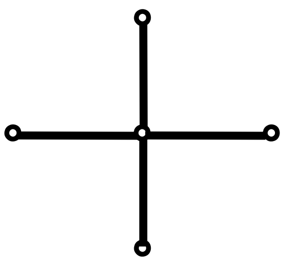
例7.2：画出5阶所有非同构的无向树。

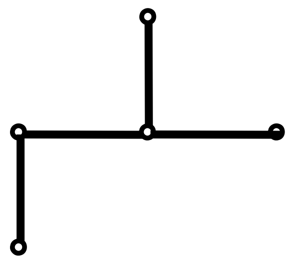
解：设T为5阶无向树，则T的边数为4， T的度序列之和为8，$\Delta(T) \leqslant 4$，$\delta(T) \geqslant 1$，可能的度序列为：

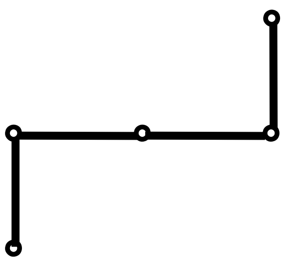
(1) 1, 1, 1, 1, 4

(2) 1, 1, 1, 2, 3

(3) 1, 1, 2, 2, 2







例7.3：无向树G有5片树叶，3个2度分支点，其余分支点均为3度，问G有多少个顶点？

解：由握手定理 $2m=\sum d(v_i)$

及定理7.1 $n = m+1$

设G有n个顶点,则有下列关系式

$$5\times 1+3\times 2+(n-5-3)\times 3=2\times (n-1)$$

解得：$n=11$

例7.4：无向树G有2个2度顶点，1个3度顶点，3个4度顶点，则其1度顶点数为多少？

解：由握手定理 $2m=\sum d(v_i)$

及定理7.1　$n = m+1$

设G有t个1度顶点, 则有下列关系式

$2 \times 2+3+4 \times 3+t =2m$

$=2\,(n-1)$

$=2\,(2+1+3+t-1)$

解得：$t = 9$

例 7.5：无向树G有8片树叶，2个3度分支点，其余分支点均为4度，问G有多少个4度分支点？画出其非同构的情况。

解：设G有t个4度分支点,则有下列关系式

$$8 \times 1+2 \times 3+ t \times 4 =2 \times(8+2+t-1)$$

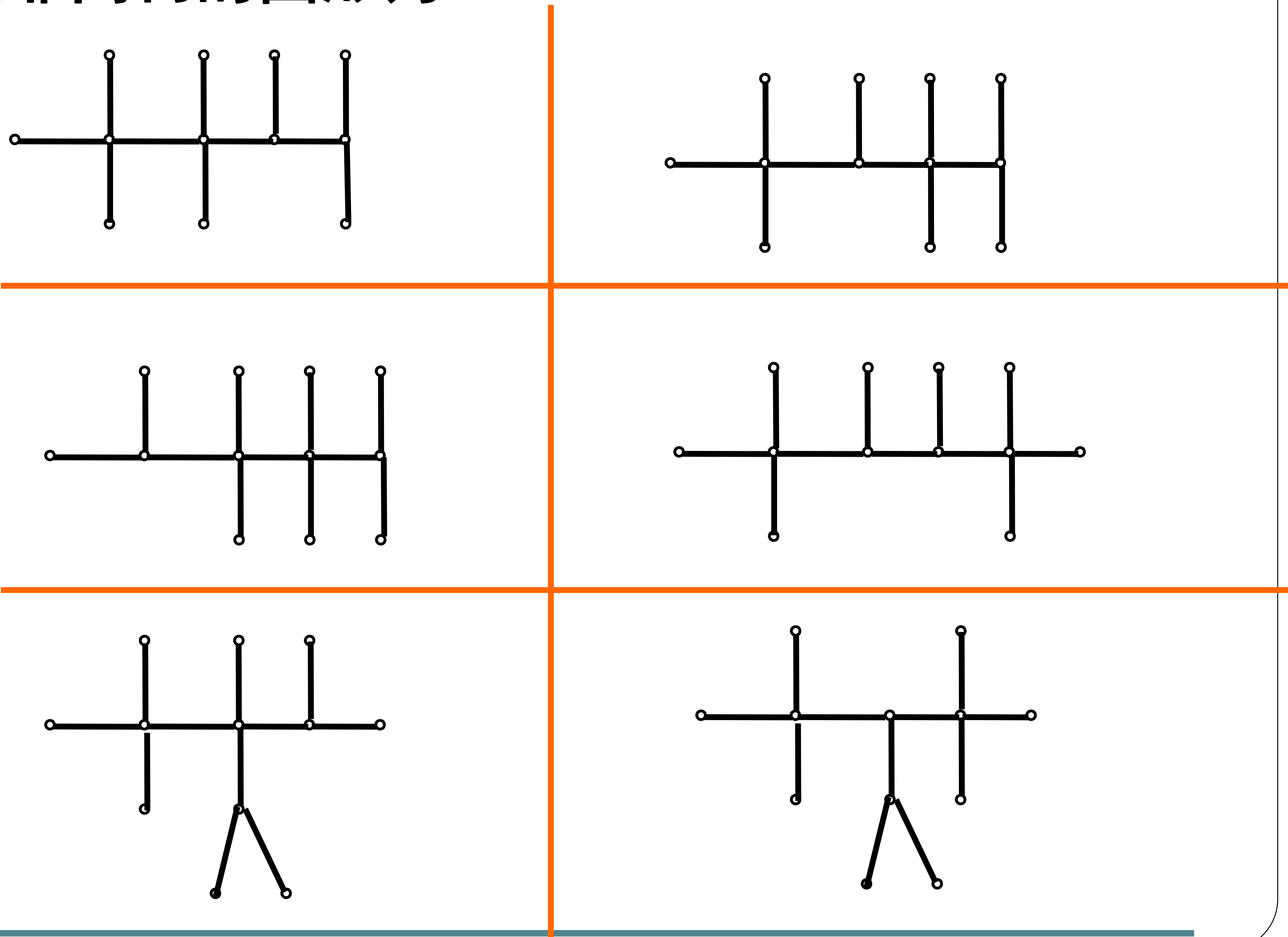
解得：$t = 2$

则G中共有12个顶点，11条边，度数序列之和为22， $\Delta(T_i)=4$，$\delta(T_i)=1$，度序列为：

1,1,1,1,1,1,1,1,3,3,4, 4

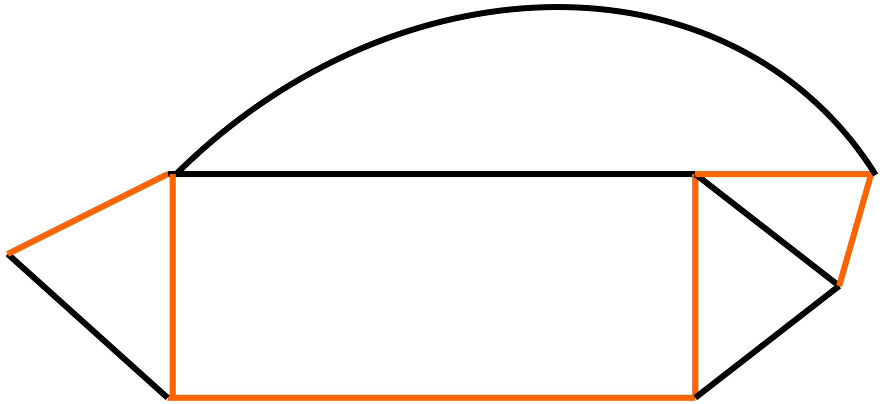
其非同构的图形为：

其非同构的图形为：



三.生成树的定义

- 设T是无向图G的生成子图并且为树，则称T为G的**生成树**。
- G在T中的边称为T的**树枝**，G不在T中的边称为T的**弦**。
- T的所有弦的集合的导出子图称为T的**余树**



橙色边表示生成树，

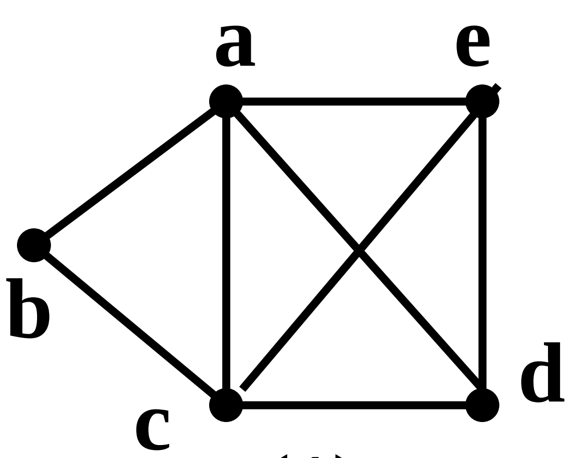
黑色边组成其余树。

余树可能不连通，

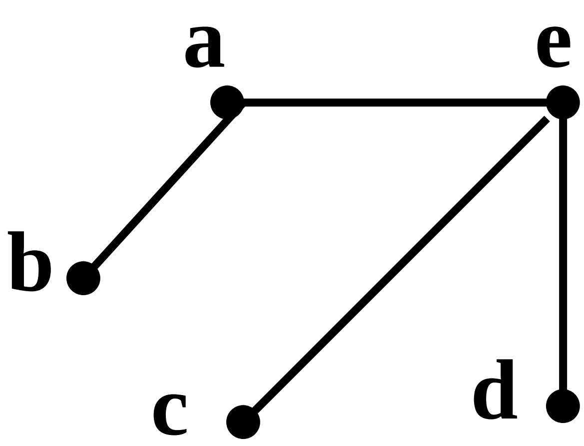
也可能含回路。

在下图中，(2)为(1)的一棵生成树T，(3)为T的余树，**注意：余树不一定是树。**

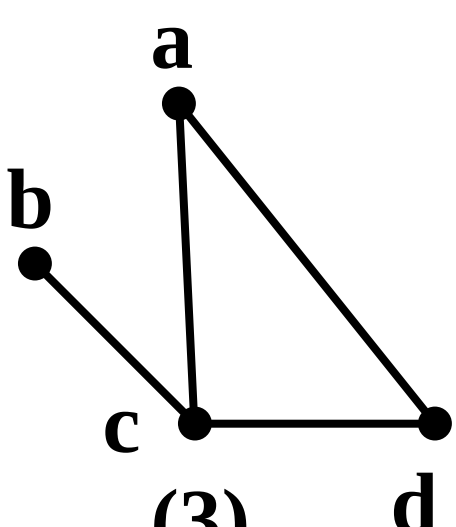
一个无向连通图，如果它本身不是树，它的生成树是不唯一的。



(1)



(2)



(3)

定理7.3 任何无向连通图G都有生成树

证明：如果G中无回路，则G为树，因此G本身就是G的生成树。

若G中含回路C,则在C中任意删去一条边不影响图的连通性。

若所得图中还有回路，重复此过程，直到所得图中无回路为止，设最后的图为T，则T为G的生成树。

推论1 设G是n阶m条边的无向连通图，则$m \geq n-1$

推论2 设G是n阶m条边的无向连通图，T为G的生成树，则T的余树T'中含有$m-n+1$条边(即T'有$m-n+1$条弦)。

四. 最小生成树

对无向图或有向图的每一条边e附加一个实数$w(e)$，称作**边e的权**. 图连同附加在边上的权称作**带权图**，记作$G=<V,E,W>$. 设T是G的生成树，T所有边的权的和称作**T的权**，记作$W(T)$.

最小生成树：G的所有生成树中权最小的生成树称为G的**最小生成树**。

- 求最小生成树的算法很多，
  我们只介绍**避圈法**（Kruskal算法）

克鲁斯卡尔算法 — 一种求最小生成树的算法

设n阶无向连通带权图G=<V,E,W>有m条边，不妨设G中无环(否则可先删去)，算法为：

(1) 将m条边按权从小到大顺序排列，设为

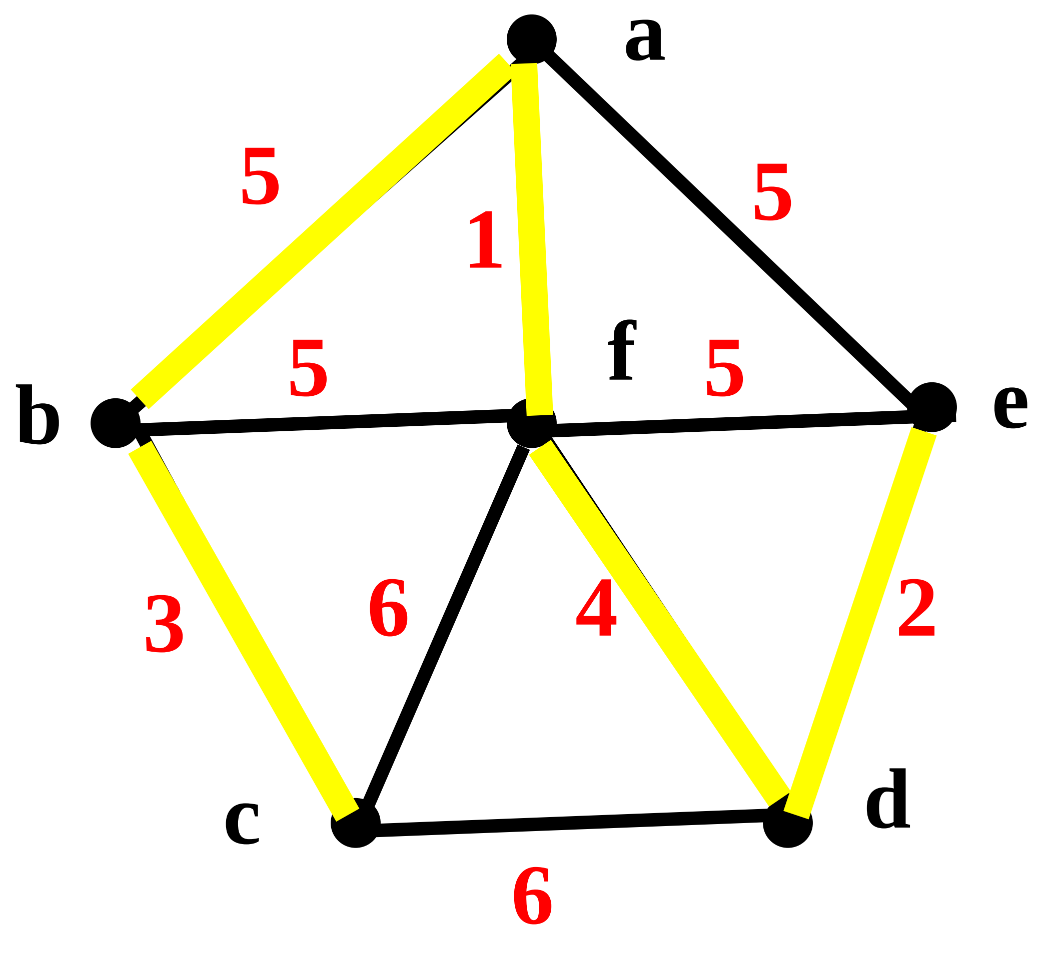
$e_1, e_2, \ldots, e_m$

(2) 取e_1在T中，然后依次检查$e_2, \ldots, e_m$，若e_j (j=2,3, …,m)与T中的边不能构成回路，则取e_j在T中，否则放弃e_j，考虑下一条边，直至j>m

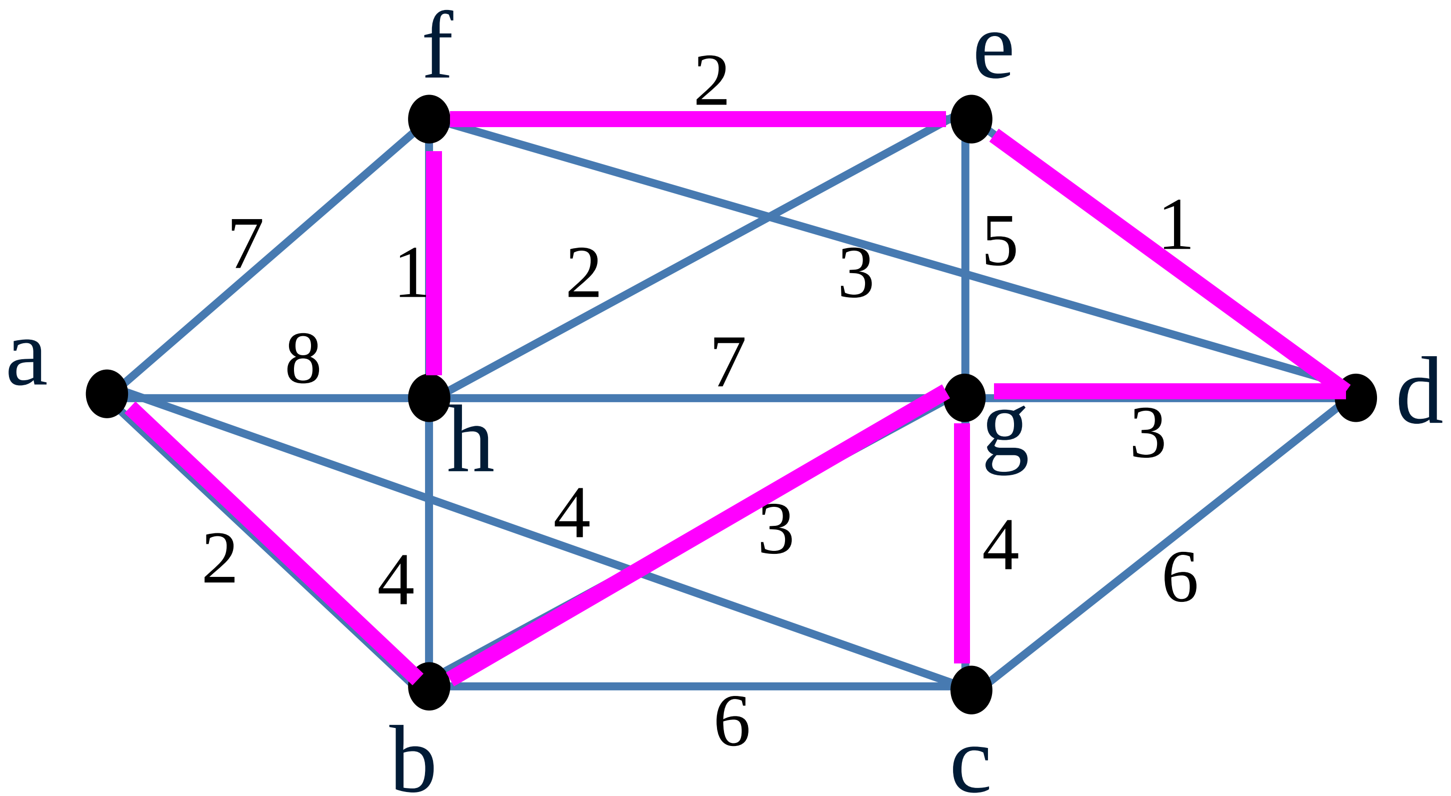
(3) 算法停止时得到的T为G的最小生成树

例7.6 用避圈法求下图所示的最小生成树

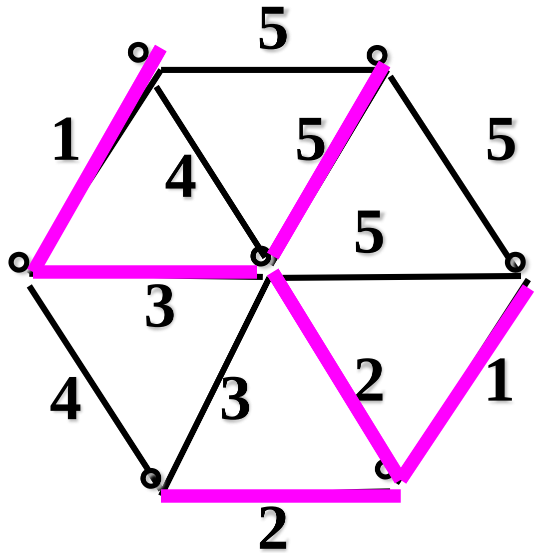
解: W(T)=1+2+3+4+5=15



例7.7 铺设一个连接各个城市的光纤通信网络

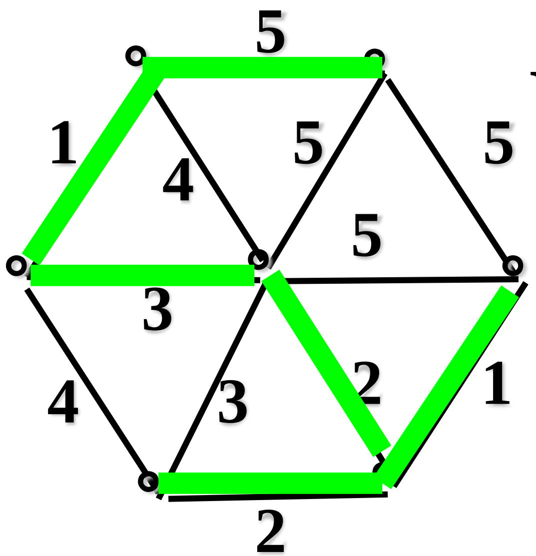


例7.8 求下图的最小生成树。



$W(T)=1+1+2+3+2+5$

$=14$



$W(T)=1+1+2+3+2+5$

$=14$

7.2 根树及其应用

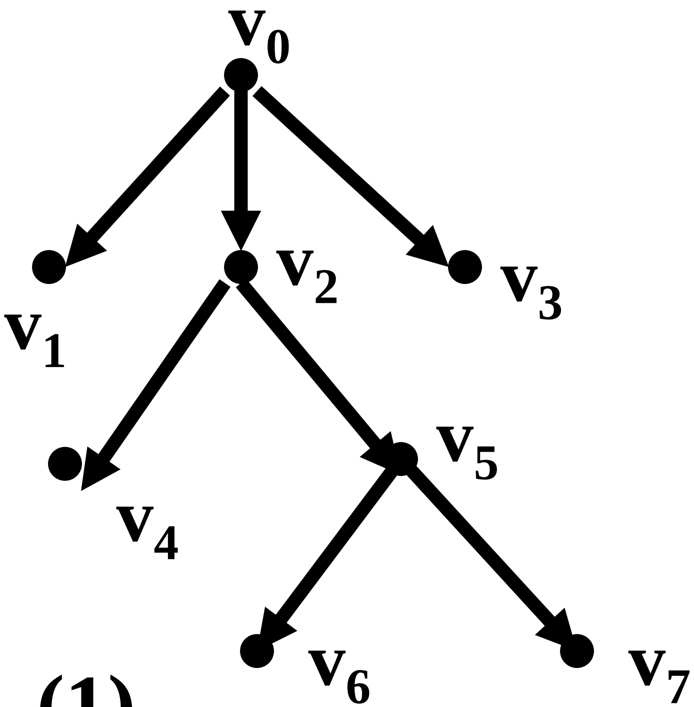
设D是有向图, 如果略去有向边的方向所得无向图为一棵无向树,则称D为有向树。

其中根树最为重要。

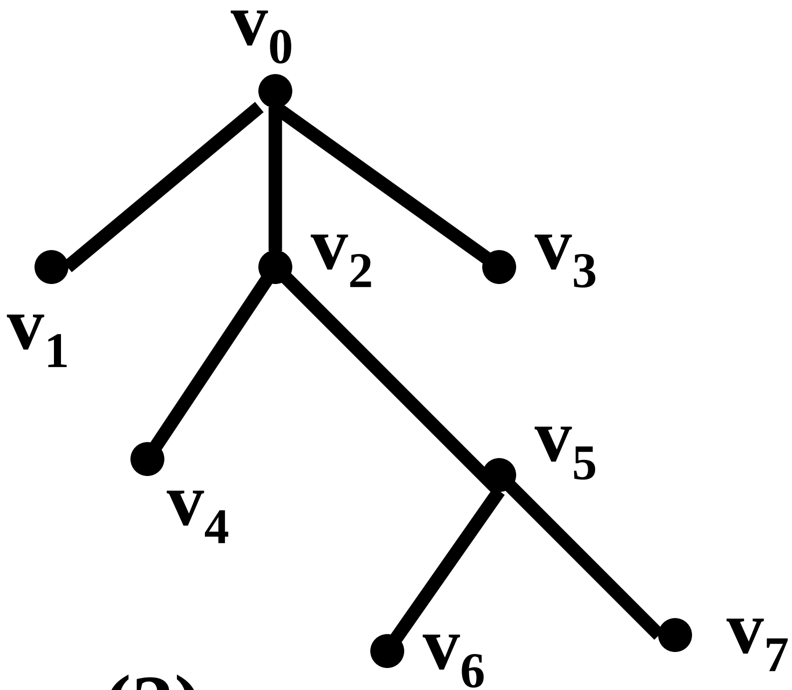
定义 7.6 根树

- 设T是$n(n \geq 2)$阶有向树，若T中有一个顶点的入度为0，其余的顶点的入度均为1，则称T为根树
- 入度为0的顶点称为树根
- 入度为1出度为0的顶点称为树叶
- 入度为1出度不为0的顶点称为内点
- 内点和树根统称为分支点

例 7.9 下图(1)为一棵根树。v_0为树根, v_1,v_4, v_3, v_6, v_7为树叶, v_2, v_5为内点, v_0, v_2, v_5为均为分支点, 由于在根树中有向边的方向均一致, 故可省掉其方向, 如图(2)



(1)



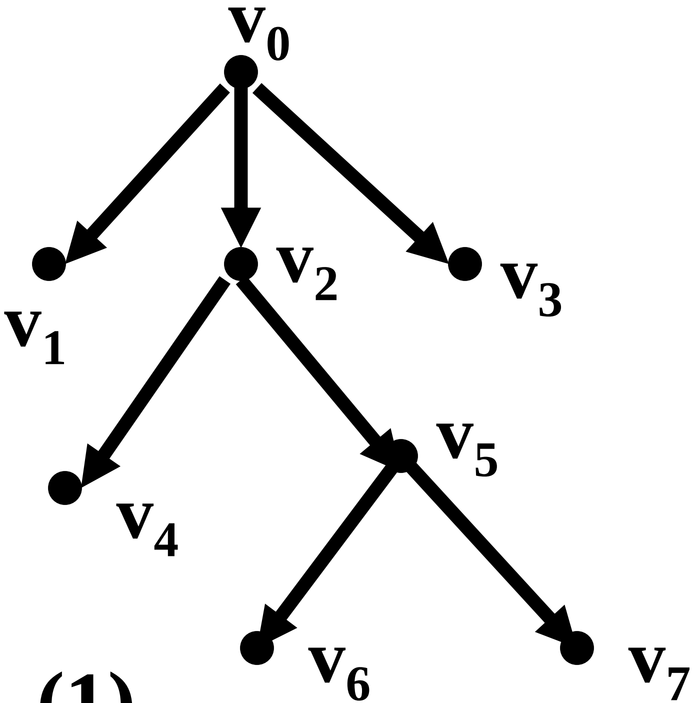
(2)

❖从树根到T的任意顶点v的通路长度称为v的层数

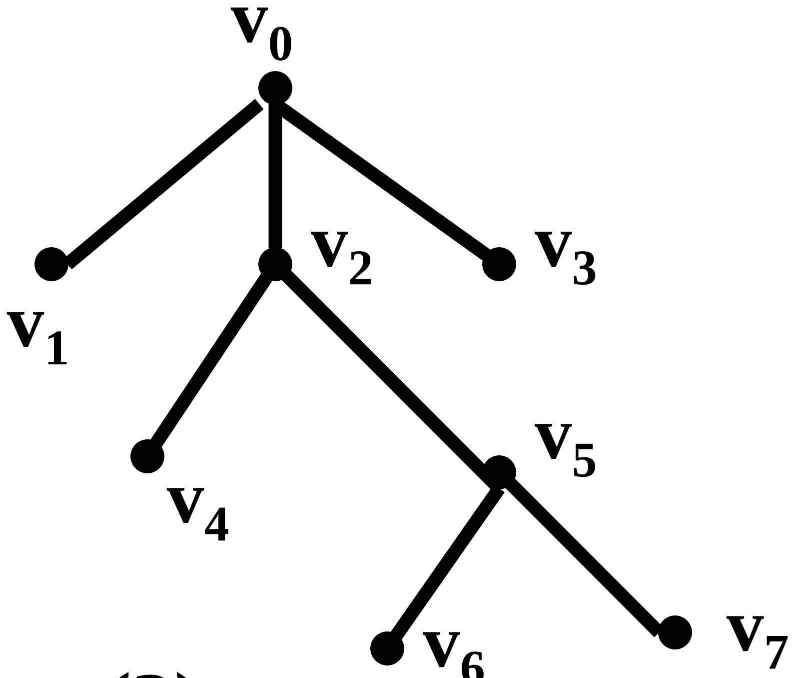
❖层数最大顶点的层数称为树高

❖平凡树也称为根树

❖例 7.9 下图(1)中, v_1,v_2, v_3,在第一层上，v_4, v_5 在第二层上，v_6, v_7在第三层上。树高为3。



(1)



(2)

定义 7.6 家族树

一棵根树可以看做一棵**家族树**

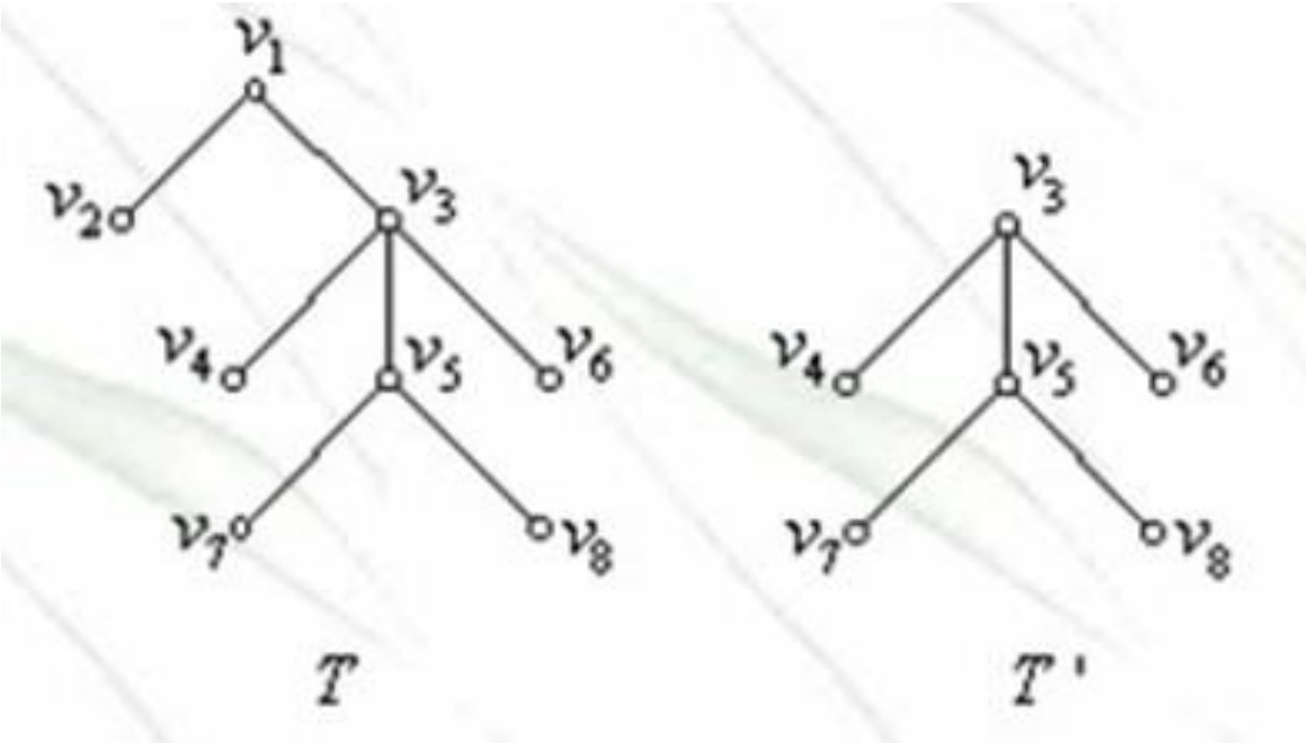
(1)若顶点 a 邻接到顶点 b，则称 b 为 a 的儿子，a 为 b 的父亲。

(2)若 b,c 的父亲相同，则称 b,c 为兄弟。

(3)若$a \neq d$,且a可达d, 则称 a 为 d 的祖先，d 为 a 的后代。

定义7.7 设T为一棵根树，a为T中的一个顶点，且a不是树根，称a及其后代导出的子图T'为T的以a为根的子树，简称**根子树**。

例7.10



根树T中，v_2, v_3为兄弟，他们为v_1 的儿子，

而v_3有儿子v_4, v_5, v_6，

v_5有儿子v_7和v_8, v_3为v_7, v_8的祖先, v_7, v_8为v_3的后代…

T'为T的根子树。

定义7.8 设T为根树，若将T中层数相同的顶点都标定次序，则称T为**有序树.**

根据根树T中每个分支点儿子数以及是否有序，可以将根树分成下列各类；

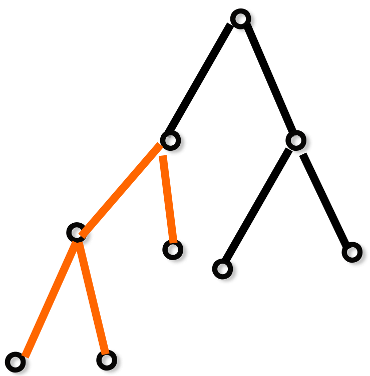
定义7.9 设T为一棵根树

(1) 若T的每个分支点至多有r个儿子，则称T为 r叉树

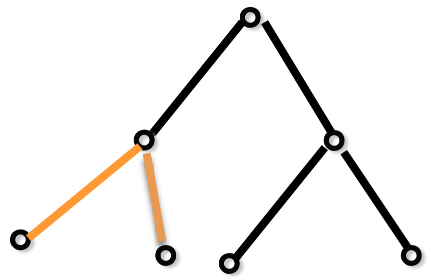
(2) 若T的每个分支点都恰好有r个儿子，为 r 元正则树

(3) 若T是r元正则树，且所有树叶的层数相同，则称T为r元完全正则树

例 7.11



二叉树(二元树)
二叉正则树



二叉完全正则树

最优2元树

定义7.10 设2元树T有t片树叶 $v_1, v_2, \ldots, v_t$, 树叶的权分别为$w_1, w_2, \ldots, w_t$,称

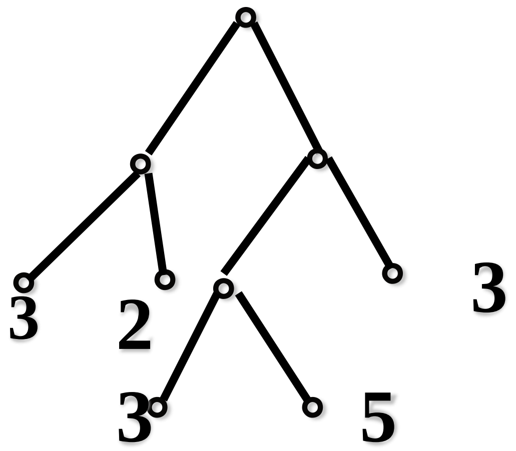
$$W(T) = \sum_{i=1}^{t} w_i l(v_i)$$

为T的权，其中 $l(v_i)$是v_i的层数.

在所有有t片树叶，带权$w_1, w_2, \ldots, w_t$的2元树中，权最小的2元树称为**最优2元树**.

例7.12 求2元树T的权。

解: $W(T)=3\times 2+2\times 2+3\times 3$
$+5\times 3+3\times 2=40$

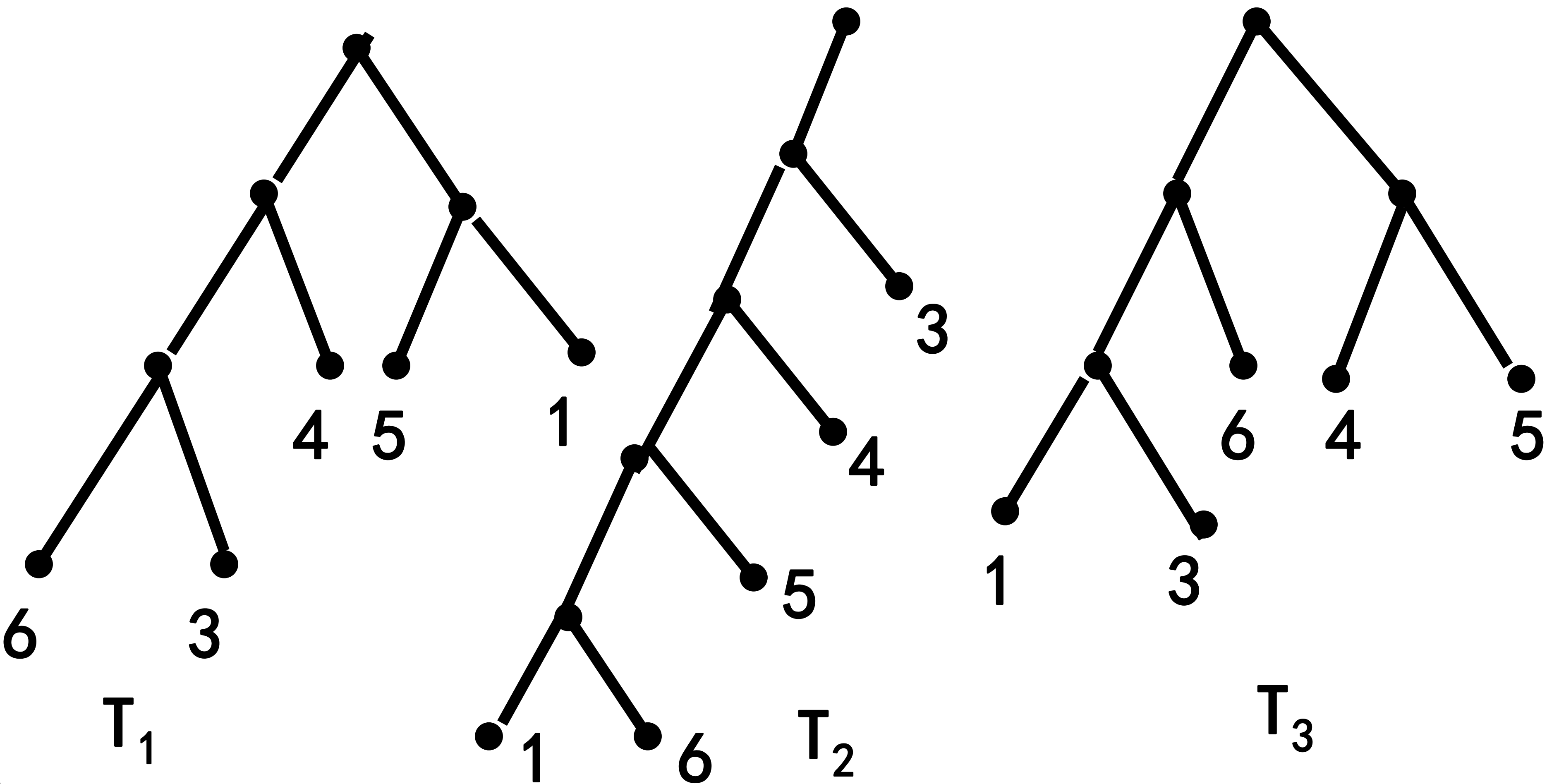


是不是最优
2元树呢？？？

例7.13下图所示三棵树都是带权1,3,4,5,6的二元树，

$W(T_1)=47, W(T_2)=54, W(T_3)=42$,

但它们中有无最优2元树还不知道



Huffman算法 ——
一种求最优二元树的算法

给定实数$w_1, w_2, \ldots, w_t$，且$w_1 \le w_2 \le \ldots \le w_t$

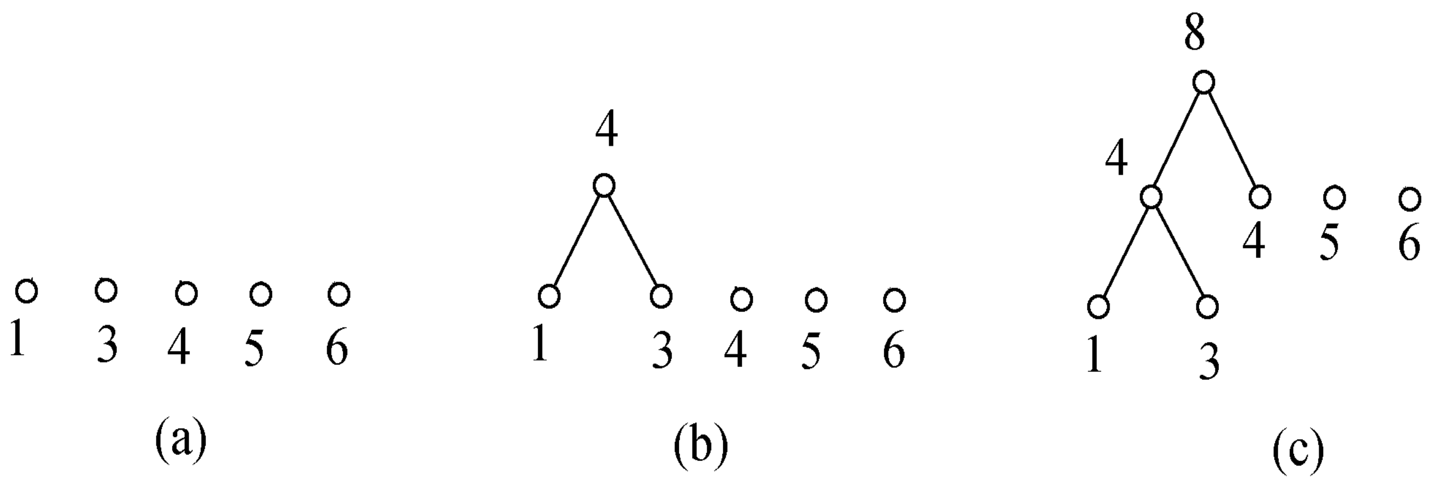
（1）连接权为w_1, w_2的两片树叶, 得一个分支点其权为w_1+w_2

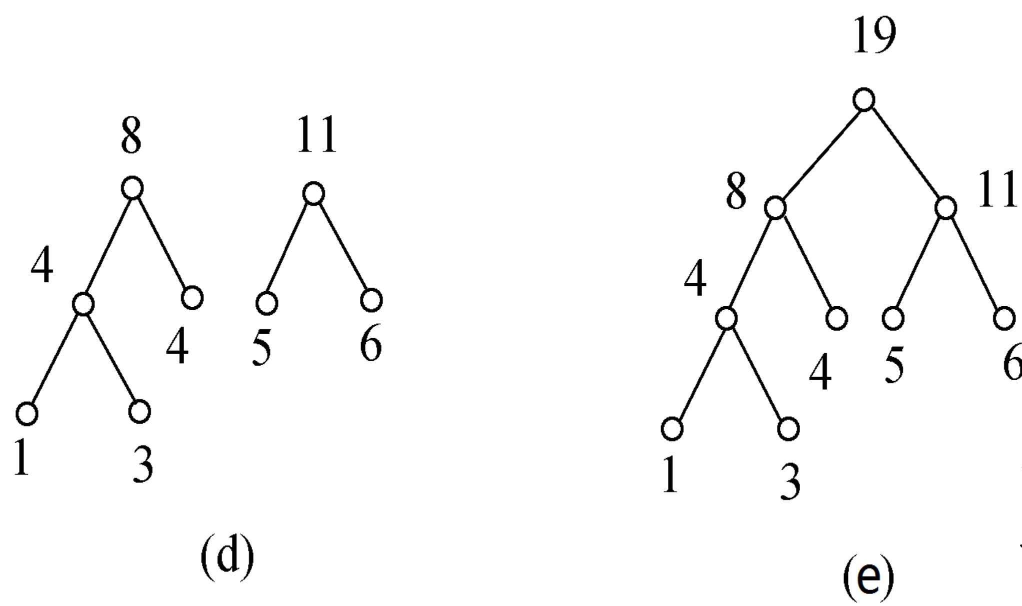
（2）在$w_1+w_2, w_3, \ldots, w_t$中选出两个最小的权, 连接它们对应的顶点(不一定是树叶), 得新分支点及所带的权。

（3）重复 (2), 直到形成t-1个分支点, t片树叶为止。

用此算法求上例的最优二叉树。

例　求权为 1，3，4，5，6的最优二元树.





$\boldsymbol{W(T)=42}$,

前面的T_3也是最优的.

例7.14 求带权1, 3, 4, 5, 6的最优二元树。

例7.15 求带权 3, 4, 5, 6, 12的最优二元树。

练习：求带权 2, 4, 7, 8, 10, 12的最优二元树。

最优二元树在通信编码中的应用

定义：前缀码

设$\alpha=\alpha_1\alpha_2\ldots\alpha_{n-1}\alpha_n$是长度为$n$的符号串

α的**前缀**: $\alpha_1\alpha_2\ldots\alpha_k$, $k=1,2,\ldots,n-1,n$

前缀码: $\{\beta_1, \beta_2, \ldots, \beta_m\}$, 其中$\beta_1, \beta_2, \ldots, \beta_m$为非空字符串, 且任何两个互不为前缀

2元前缀码: 只有两个符号(如0与1)的前缀码

如 {0,10,110, 1111}, {10,01,001,110}是2元前缀码

{0,10,010, 1010} 不是前缀码

如何产生二元前缀码?

用二叉树产生二元前缀码之做法

给定一棵2叉树T，设它有t片树叶。设v为T的一个分支点，则v至少有一个儿子，最多有两个儿子。

若v有两个儿子，在由v引出的两条边上，左边的标上0，右边的标上1；

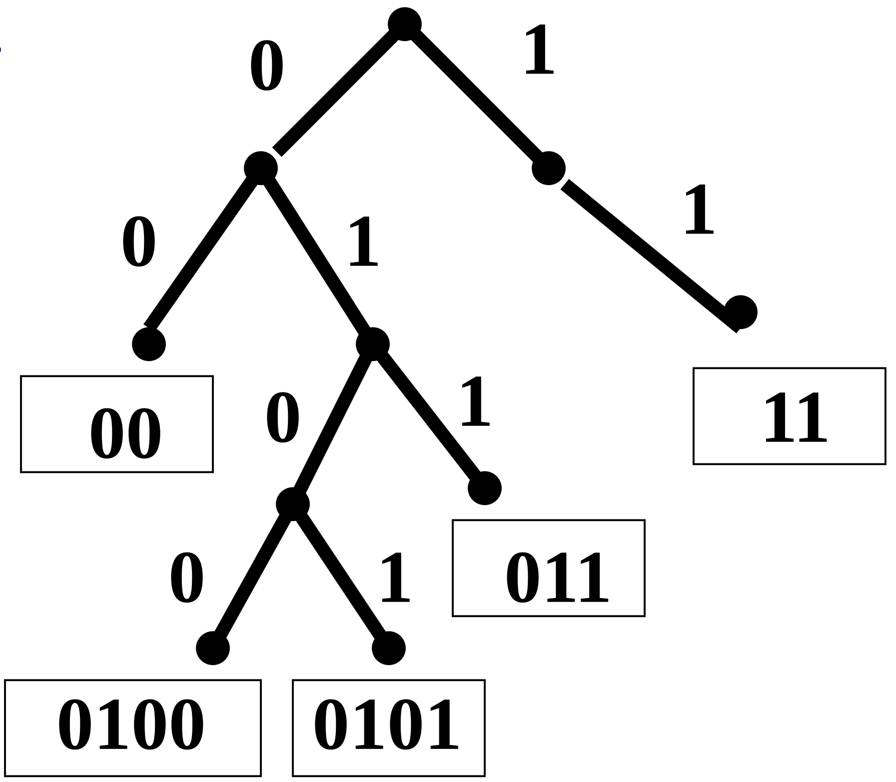
若v有一个儿子，在由v引出的边上可标上0，也可标上1。

设v_i为T的任一片树叶，从树根到v_i的通路上各边的标号组成的0，1串组成的符号串放在v_i处，t片树叶处的t个符号串组成的集合为一个二元前缀码。

例7.16

右图所示的二元树产生的前缀码为

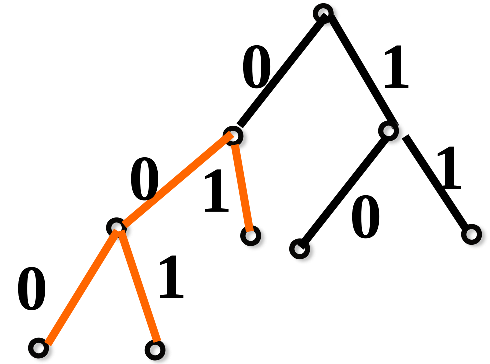
{11,00,011,0100,0101}



由一棵给定的2叉正则树，可以产生唯一的一个二元前缀码。

例7.17

右图所示的是一棵2叉正则树，它产生唯一的一个二元前缀码是{000, 001, 01, 10, 11}



应用：传输按着一定比例出现的符号时，要求寻找传输它们**最省二进制数字**的前缀码—**最佳前缀码**。

设要传输的电文中含有t个字符, 字符a_i出现的频率为p_i, 它的编码的长度为l_i, 那么100个字符的电文的编码的期望长度是$100\sum_{i=1}^{t} l_i p_i$.

称编码期望长度最小的2元前缀码为**最佳2元前缀码.**

在用2叉树产生2元前缀码时, 每个二进制串的长度等于它所在树叶的深度, 因而权为$100p_1$, $100p_2$, …, $100p_t$的最优2叉树产生的2元前缀码是最佳2元前缀码. 于是, 给定字符出现的频率, 可以用Huffman算法产生最佳2元前缀码.

例7.18　在通信中，0, 1, 2, 3, 4, 5, 6, 7出现的频率如下

0：30%，　1：20%，　2：15%

3：10%，　4：10%，　5：5%

6：5%，　7：5%

求传输它们的最佳前缀码.

例7.18 在通信中, 0,1,2,3,4,5,6,7出现的频率如下

0：30%， 1：20%， 2： 15%

3：10%， 4：10%， 5： 5%

6： 5%， 7：5%

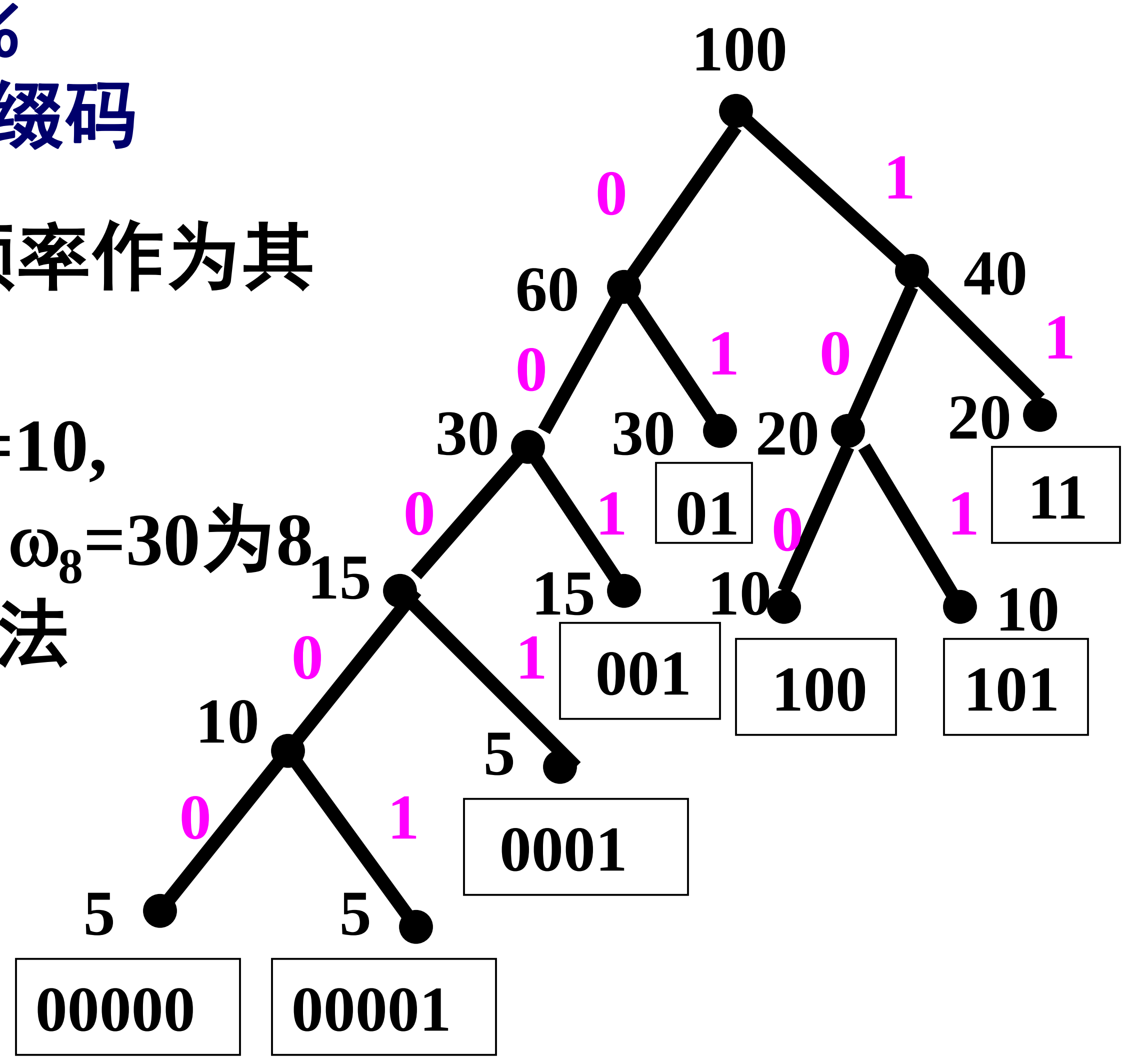
求传输它们的最佳前缀码

解:将各字符出现的频率作为其相应的权

$\omega_1=5$, $\omega_2=5$, $\omega_3=5$, $\omega_4=10$, $\omega_5=10$, $\omega_6=15$, $\omega_7=20$, $\omega_8=30$为8个权,利用*Huffman*算法求出的

最优2叉树如图所示

(码长取3,如101传5)



例7.18 在通信中, 0,1,2,3,4,5,6,7出现的频率如下

0：30%， 1：20%， 2： 15% 3：10%，

4：10%， 5： 5% 6： 5%， 7：5%

求传输它们的最佳前缀码

图中方框中的8个码子是**最佳前缀码。**

树T是带权$\omega_1, \omega_2, \ldots, \omega_8$的最优二元树。

带权为ω_i的树叶v_i对应的码子传输出现频率为ω_i%, 的数字，即

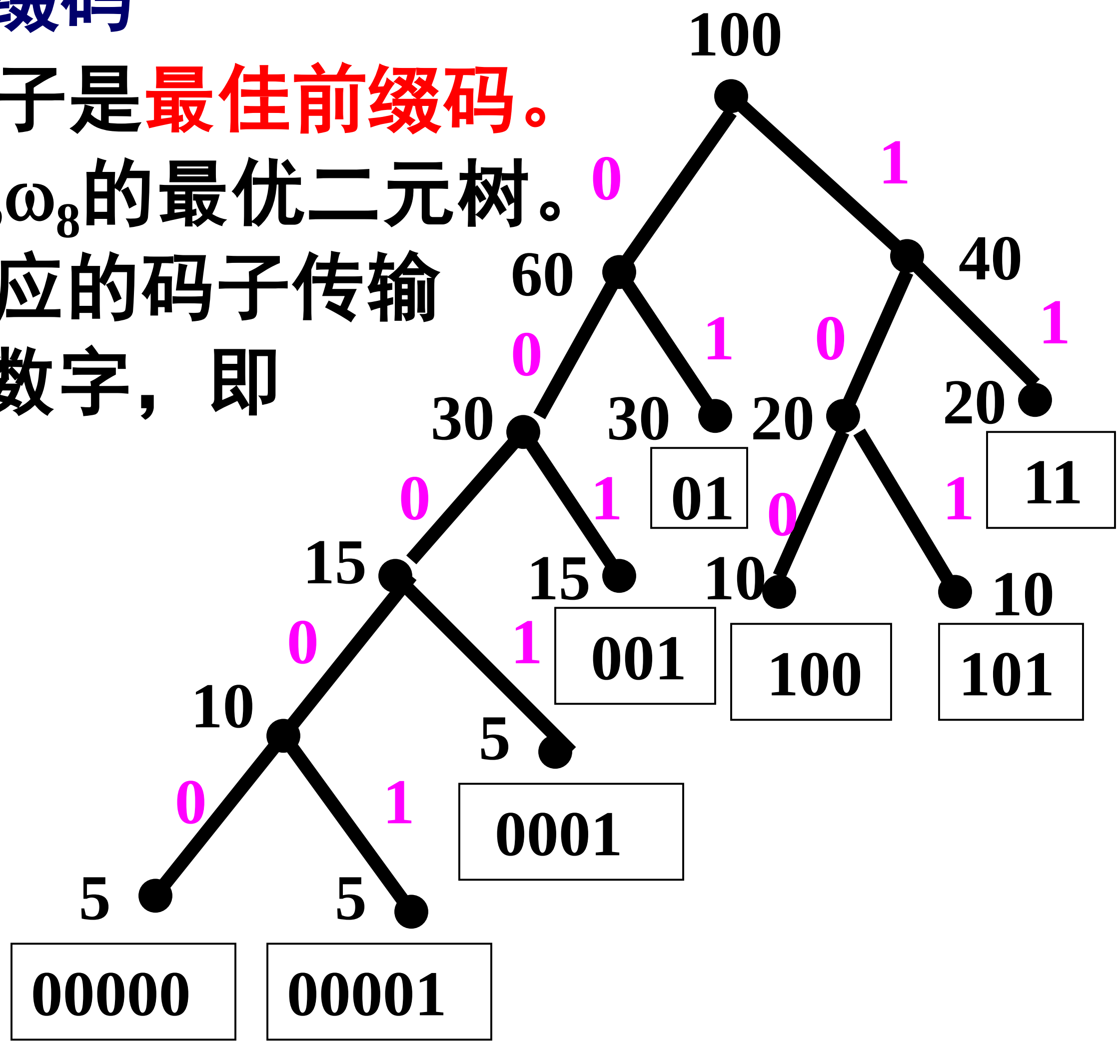
11传1， 101传4

01传0， 0001传5

001传2， 101传4

11传1， 00001传6

100传3， 00000传7



本课程到此结束

THANKS