第七章 树及其应用

郑征

zhengz@buaa.edu.cn

软件与控制研究室

第七章 树及其应用

第1讲 无向树及生成树

半决赛

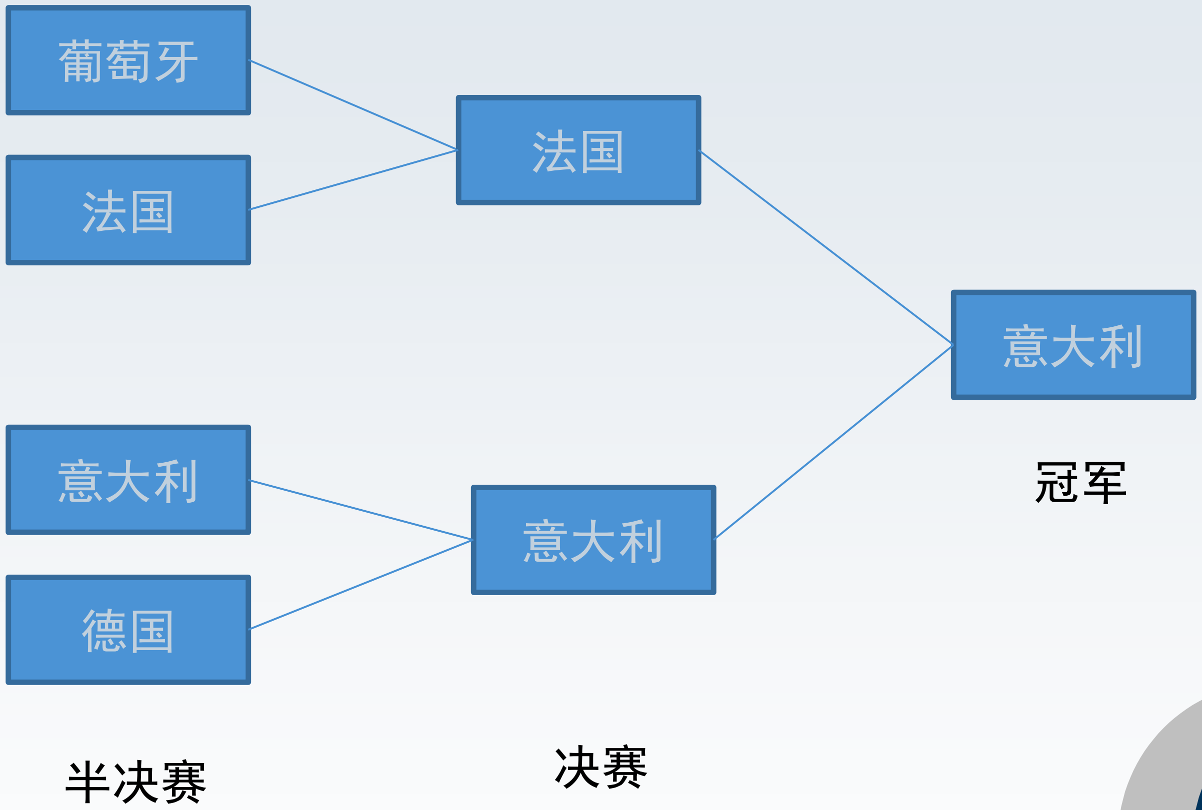
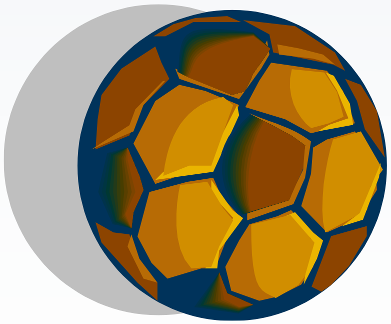
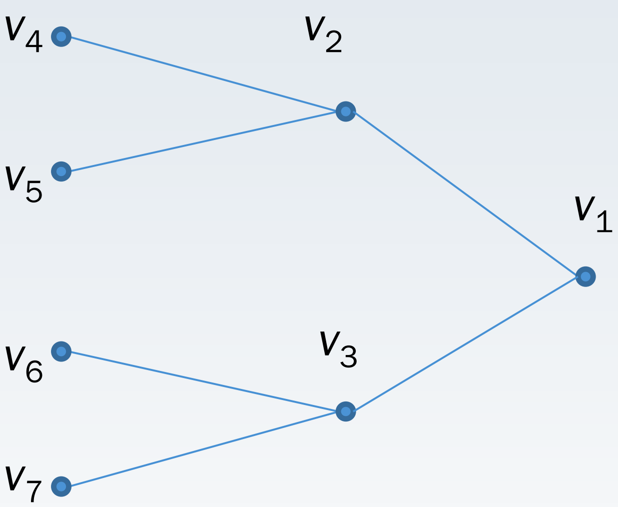


图7.1 06世界杯比赛的半决赛和决赛





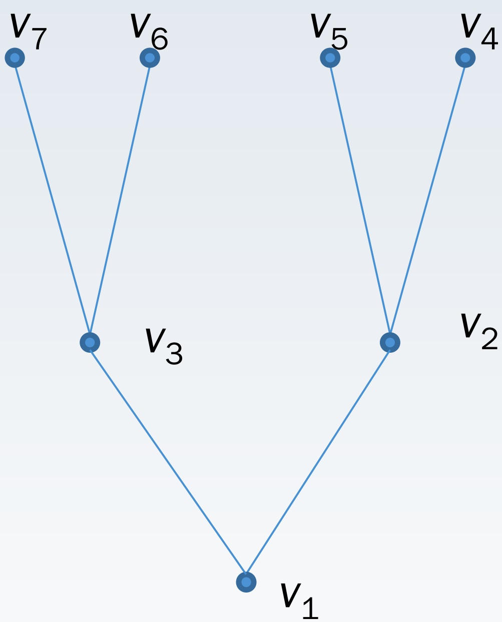
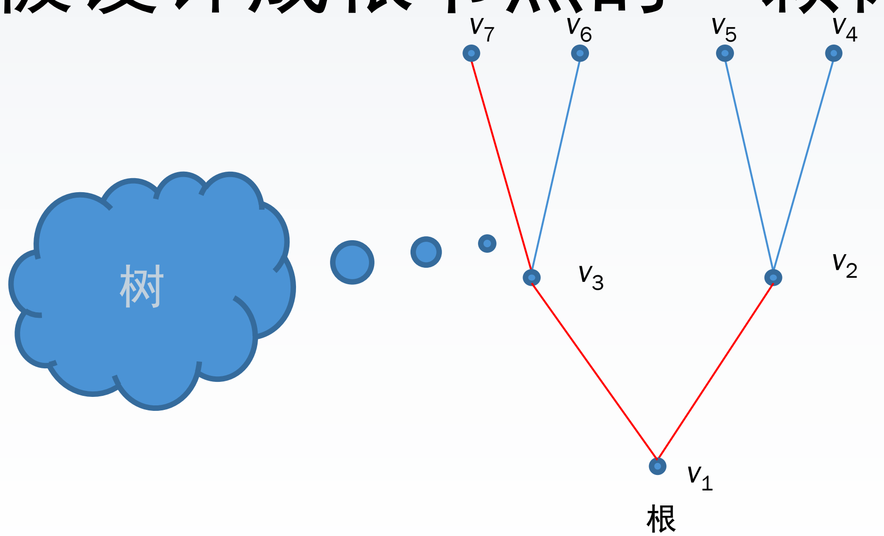


图7.2 图7.1的06世界杯比赛的树形表示

图7.3 图7.1的树旋转后与一颗自然树的比较

树的形式化定义

- 定义：一棵（自由）树T是一个简单图，满足以下条件：如果*v*和*w*是树*T*的顶点，则在*v*和*w*之间有唯一一条路径。一棵有根树就是有一个特殊的顶点被设计成根节点的一颗树。



树的应用举例

- 关系的描述——表示层次关系。例如行政组织表，计算机文件系统。
- 数据的存储——存储各类数据，可以节省空间和加快搜索。例如数据库中记录之间的逻辑关系，Huffman编码。
- 数据的精简——对大量的数据进行精简，以符合某种需要。例如，决策树，规则树等。

树的应用举例

例7.1：有根树通常会用来表示层次关系

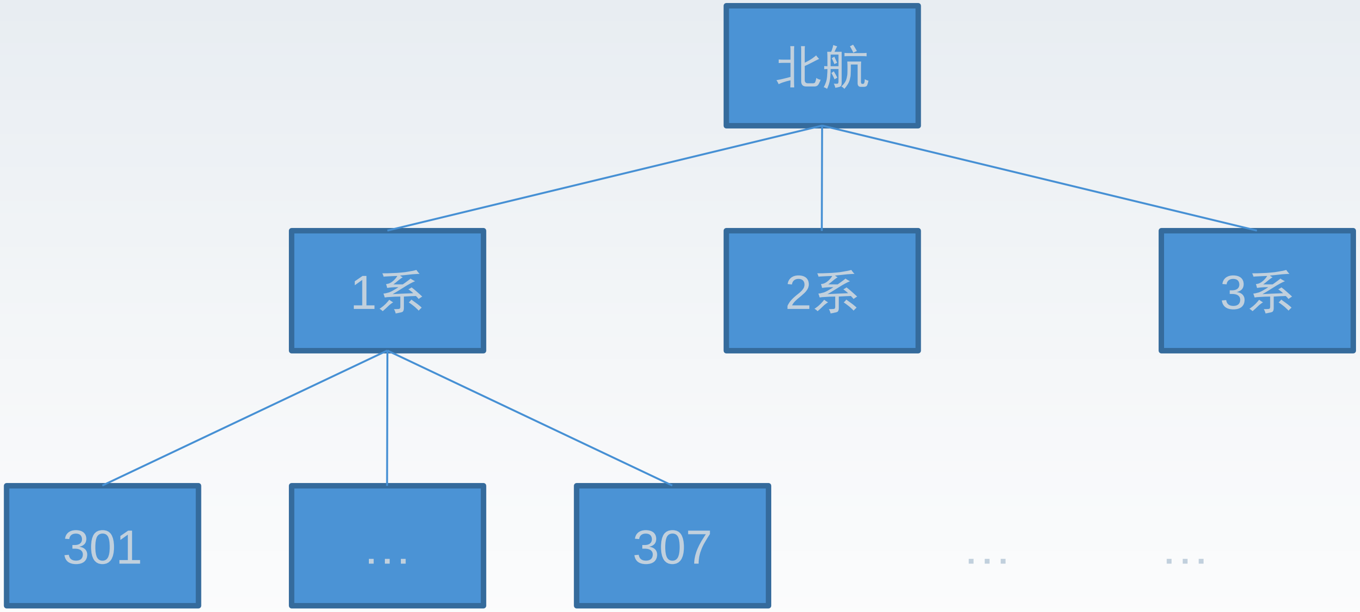


图7.6 北航行政组织架构

树的应用举例

例7.2：计算机文件系统

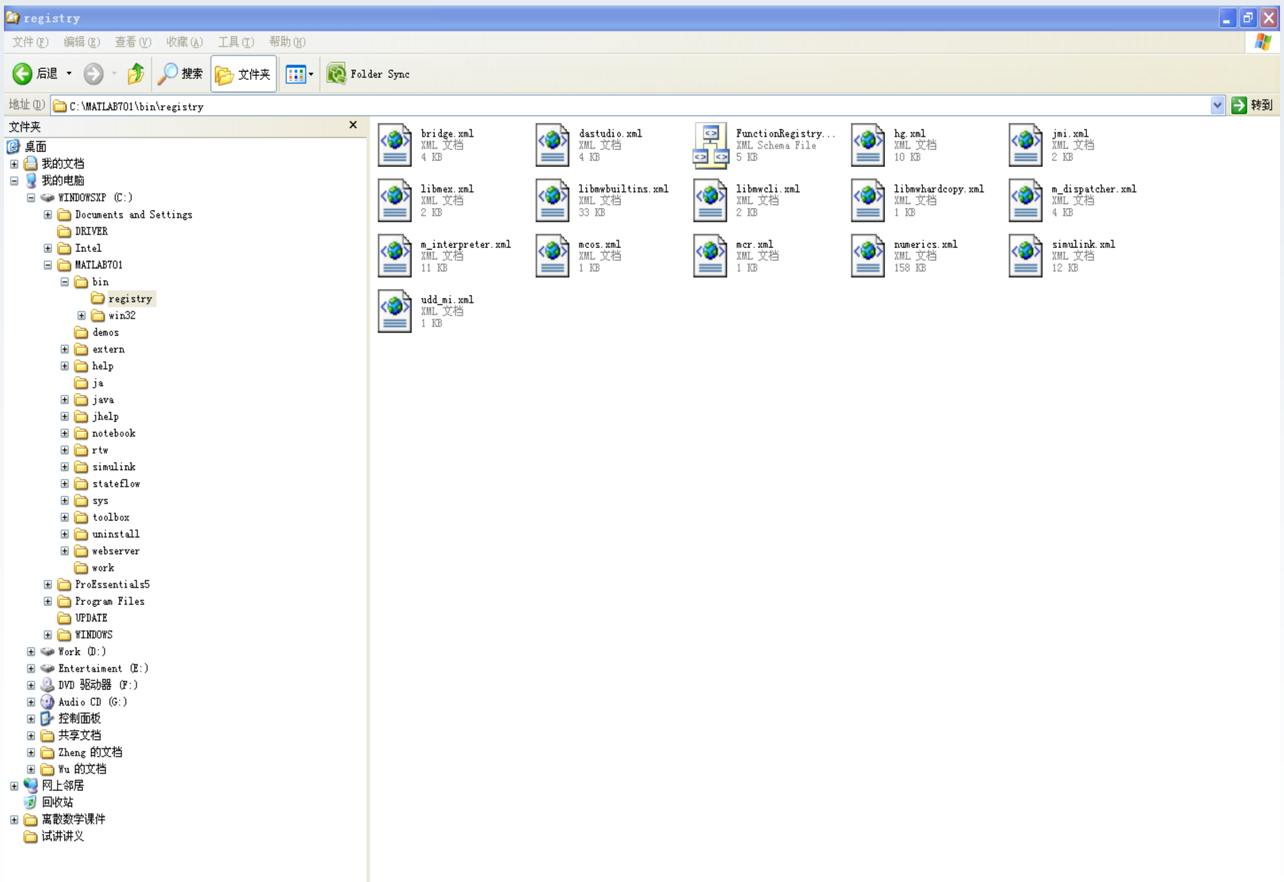


图7.7 计算机文件系统

树的应用举例

例7.2：计算机文件系统

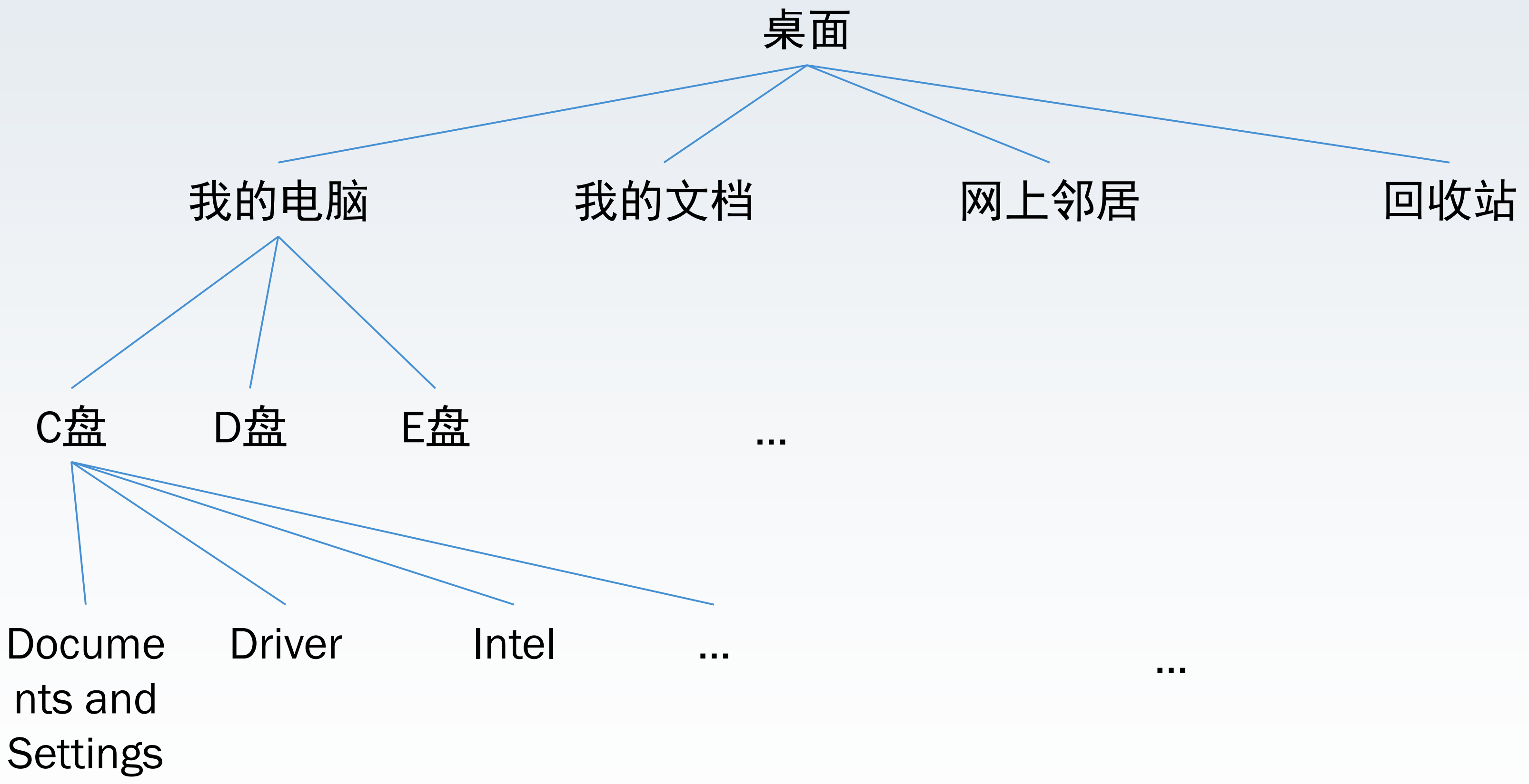


图7.8 计算机文件系统的树形表示

树的一些特性

- 连通
- 多加一条边则不是树
- 减少一条边也不是树
- 便于搜索和遍历，由于无回路，可以避免重复

无向树

* 定义 7.1
* 连通无回路的无向图称为**无向树**，简称**树**，常用T表示树。

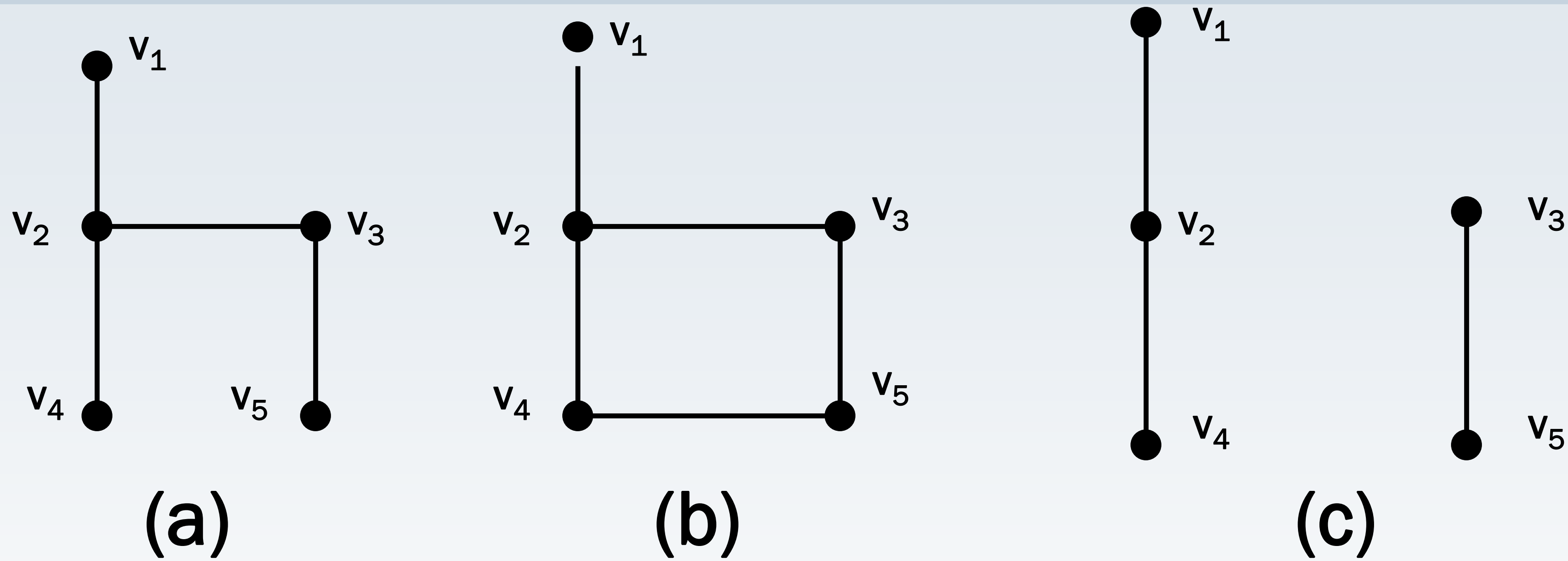
（即**树是不包含回路的连通图**）

平凡图称为**平凡树**。

若无向图G至少有两个连通分支，则称G为**森林**。

在无向树中，悬挂顶点称为**树叶**，度数大于或等于2的顶点称为**分支点**。

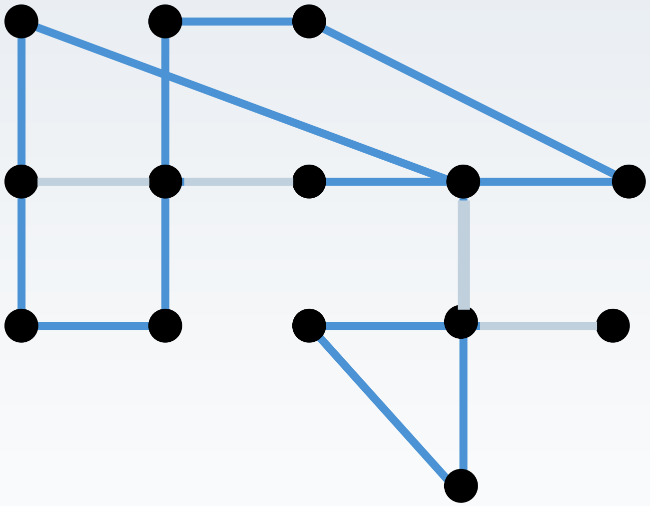
例 判断下列哪些图是树?



解: 图(a)是树, 因为它连通又不包含回路。图(b), (c)不是树, 因为图(b)虽连通但有回路, 图(c)虽无回路但不连通。

在图(a)中, v_1、v_4、v_5为均为叶, v_2、v_3均为分支节点。

例 连通图、树和森林之间的相互转换。



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个含新边的圈。

(6) G是连通的且G中任何边均为桥。

(7) G是连通的,但删除任何一条边后，就不连通了。

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：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k片树叶, 则有(n-k)个顶点度数大于等于2,由握手定理及定理7.1可知

$$2m=\sum d(v_i) \geqslant k+2(n-k)$$

由定理7.1可知, m=n-1, 将此结果代入上式后解得 $k \geqslant 2$.

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：画出5阶所有非同构的无向树。

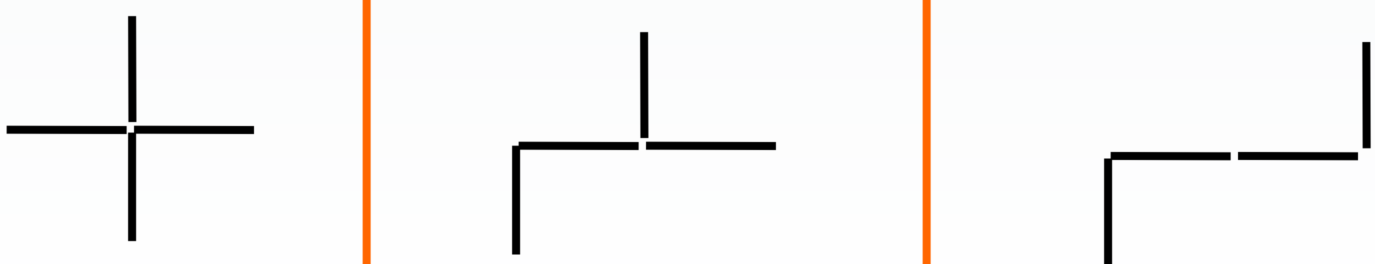
解：设T_i为5阶无向树,则T_i的边数为4， T_i的度序列之和为8， $\Delta(T_i) \le 4$, $\delta(T_i) \geq 1$, 可能的度序列为：

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

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

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

称只有一个分支点且其度数为n-1的n阶无向树为**星形图**，称唯一的分支点为**星心**。



例：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：无向树G有2个2度结点，1个3度结点，3个4度结点，则其1度结点数为多少？

解：由握手定理 $2m=\sum d(v_i)$

及定理1　$n = m+1$

设G有t个1顶点, 则有下列关系式

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

$$= 2 \times (n-1)$$

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

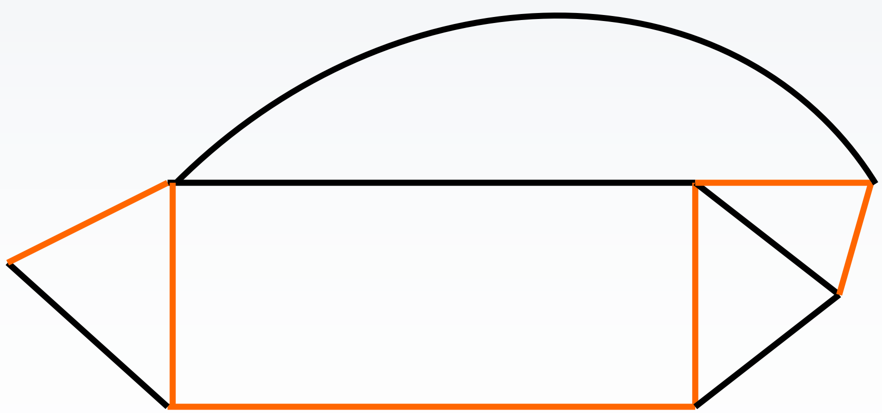
解得：$t = 9$

定义7.2

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。

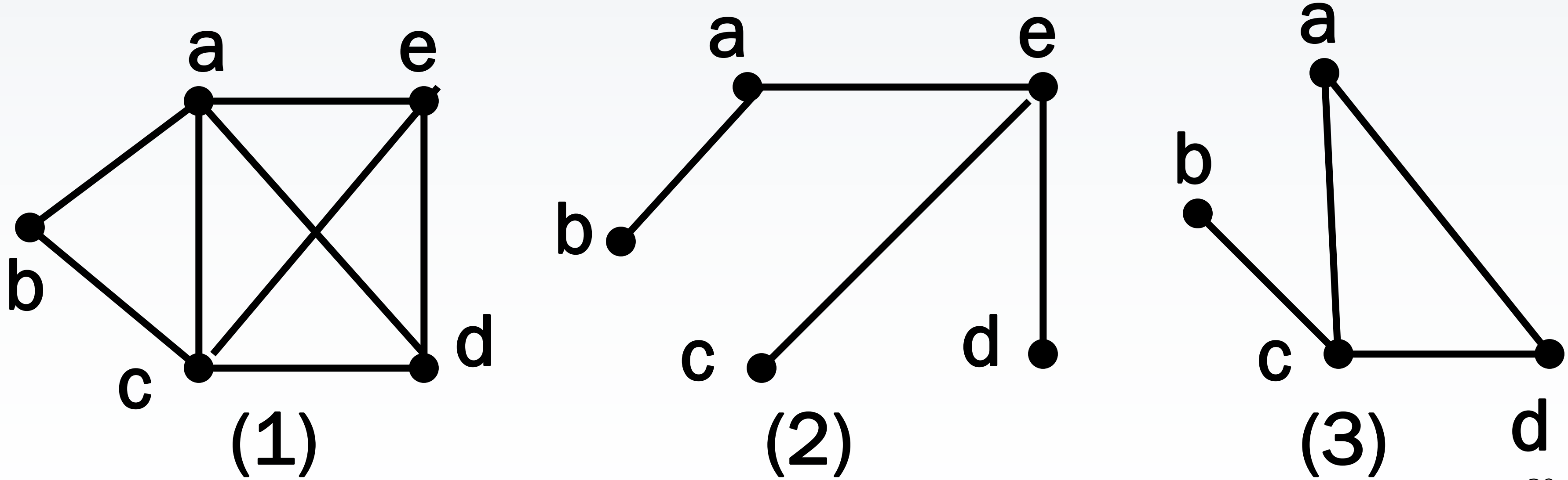
T的所有弦的集合的导出子图称为T的余树



图中，红色边表示生成树，黑色边组成其余树。可见，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。但所有的连通图都具有生成树。事实上，若G是连通图，又G中无回路，则G本身就是树。

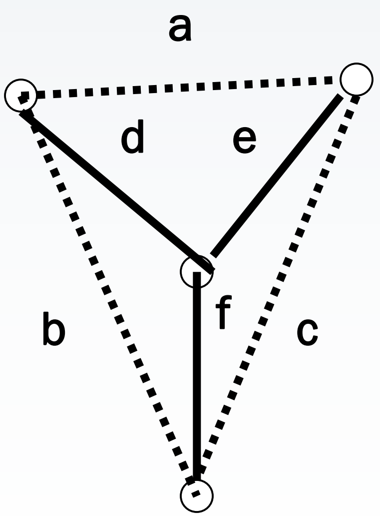


定理7.3 无向图G具有生成树⇔ G是连通图。

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边(即T'有m-n+1条弦)。

在右图中，实边所示的子图是图G的一棵生成树T，d,e,f为T的树枝，a,b,c为T的弦，在弦上加弦a，产生G的一个初级回路aed，还可在T上分别加弦b,c又可形成2个初级回路bdf和cef，这3个回路中的每个回路都只含一条弦，

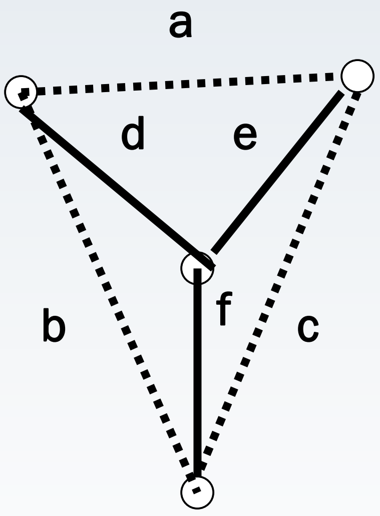


定义7.3

设T是n阶m条边的无向连通图G的一棵生成树，设$e_1,e_2, \ldots ,e_{m-n+1}$为T的弦，设$C_r$为T添加弦$e_r$产生的G的回路， $r=1,2, \ldots , m-n+1$。则称C_r为对应于弦e_r的**基本回路，**称$\{C_1,C_2,C_{m-n+1}\}$为对应生成树T的**基本回路系统**，

在右图中, C_a=aed, C_b=dbf, C_c=cef,为对应生成树T的基本回路, $\{C_a,C_b,C_c\}$为T的**基本回路系统**。

一个连通图G对应不同的生成树的基本回路及基本回路系统可能不同，但基本回路的个数G所固有的**参数**(弦)，等于m-n+1。

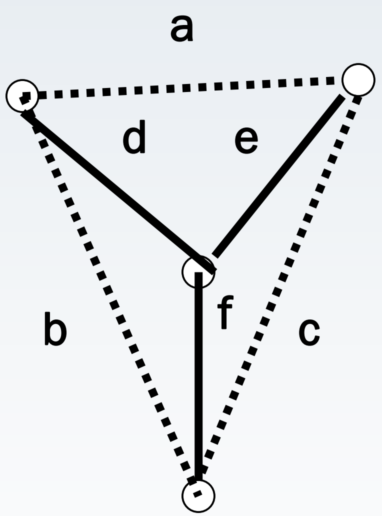


定义 7.4

设T是n阶连通图G的一棵生成树，称T的n-1个树枝对应的G的n-1个割集（每个割集只含一个树枝，其余的边都是弦）$S_1, S_2, \ldots, S_{n-1}$为对应生成树T的G的**基本割集**，称$\{S_1, S_2, \ldots, S_{n-1}\}$为对应生成树T的**基本割集系统**。

在右图中, T的树枝d对应G的一个割集{d,b,a}, e对应一个割集{e,a,c}, 树枝f对应一个割集{f,c,b}。

3个树枝对应的G的割集的特点是：每个割集中只含一个树枝，其余的边都是弦。这样的割集称为基本割集。

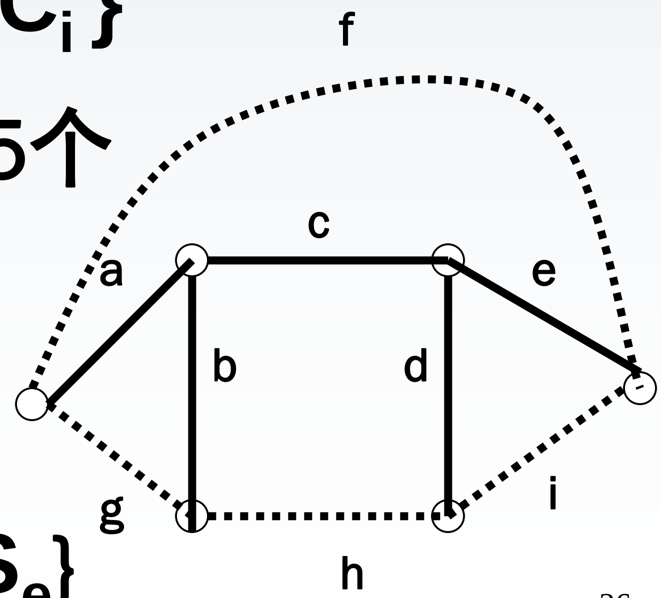


例 在右下图所示的图G中，实数边所构成的子图是G的一棵生成树T，求T对应的基本回路和基本回路系统，基本割集和基本割集系统

解：G中顶点数n=6，边数m=9，基本回路个数为m−n+1=4，即T有4条弦，f,g,h,i。对应每条弦有一个基本回路：

C_f=face; C_r=gba; C_h=hdcb; C_i=ied;

基本回路系统为{ C_f ,C_r ,C_h , C_i }



T有5个树枝a, b, c, d, e, 因而有5个基本割集:S_a={a,g,f } ;
S_b={b,g,h } ;S_c={c,f,h } ;
S_d={d,i,h } ; S_e={e,f,i}
基本割集系统为{S_a,S_b,S_c, S_d,S_e}

定义7.5

设无向连通带权图G=<V,E,W>,T是G的一棵生成树，T的各边权之和称为T的**权**，记作W(T)。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）

Kruskal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 $e_1,e_2, \dots ,e_m$。

(2) 取e_1在T中，然后依次检查$e_2, \dots ,e_m$，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 用避圈法求下图所示的最小生成树

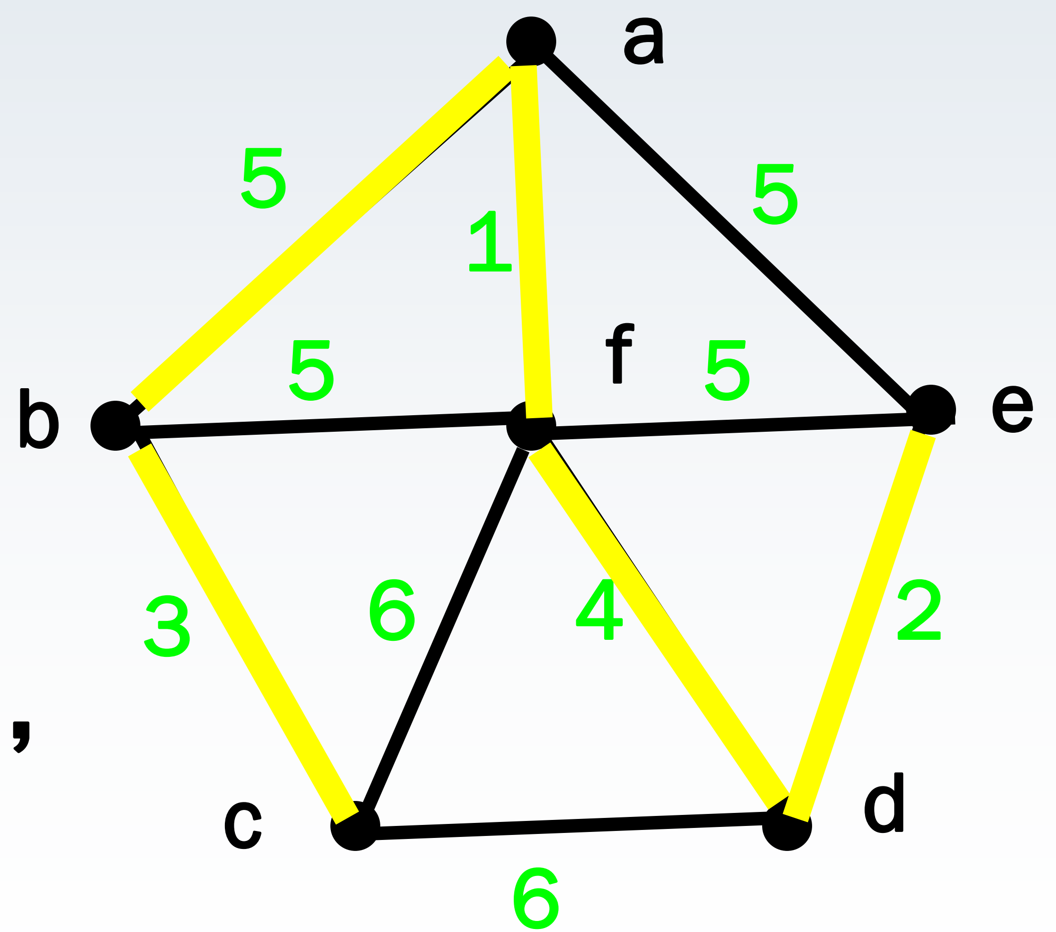
解:

W(T)=1+2+3+4+5

=15

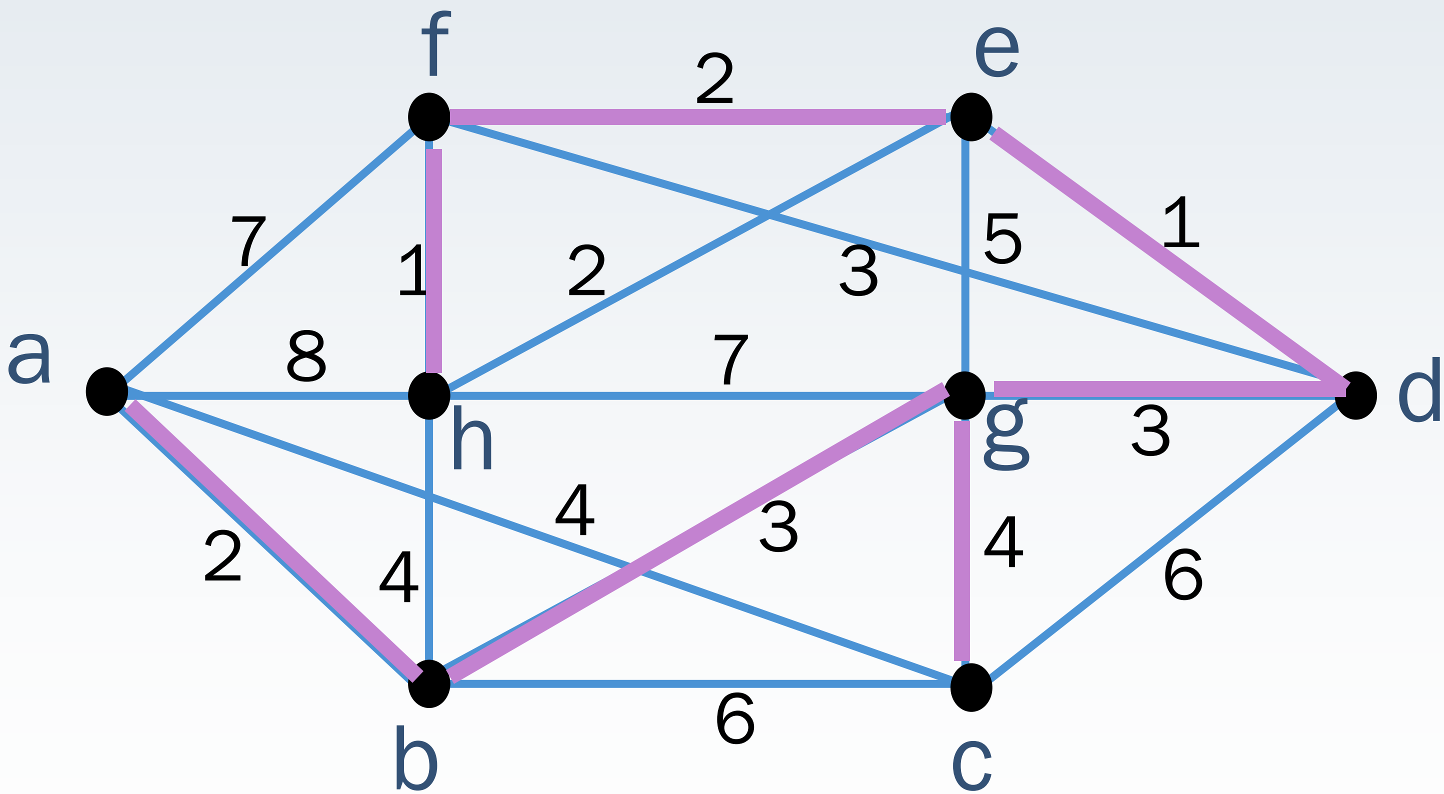
注意:

最小生成树的结点数与原图相等，边的数目比原图少。

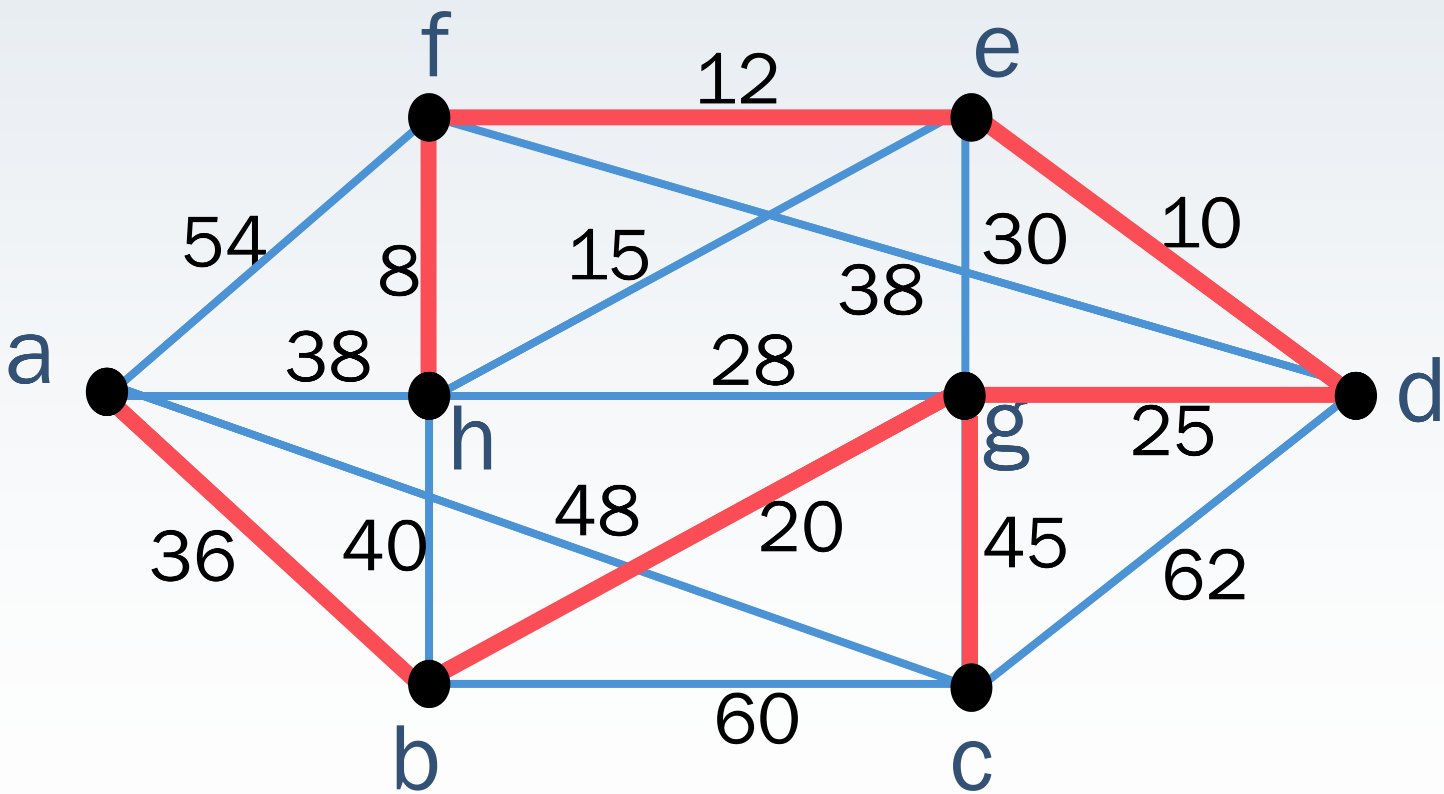


避圈法　Kruskal(克鲁斯卡尔算法)

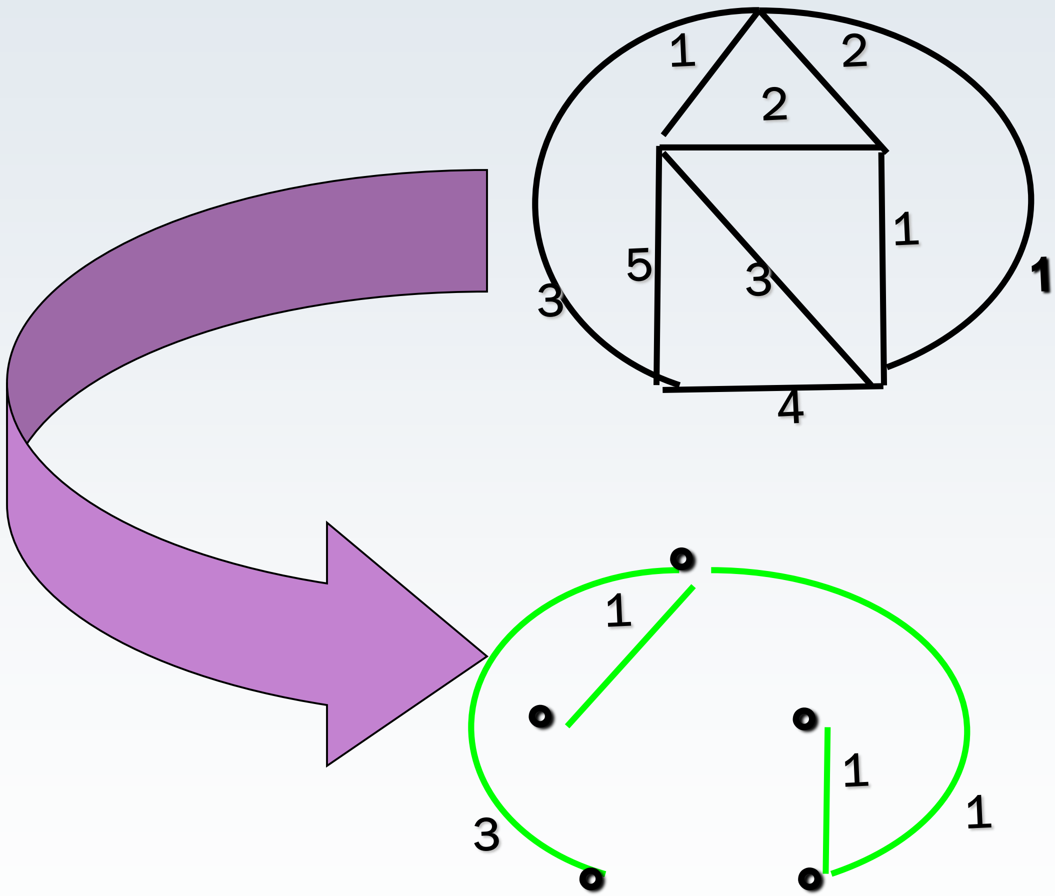
例：铺设一个连接各个城市的光纤通信网络



例：铺设一个连接各个城市的光纤通信网络。　　　（单位：万元）

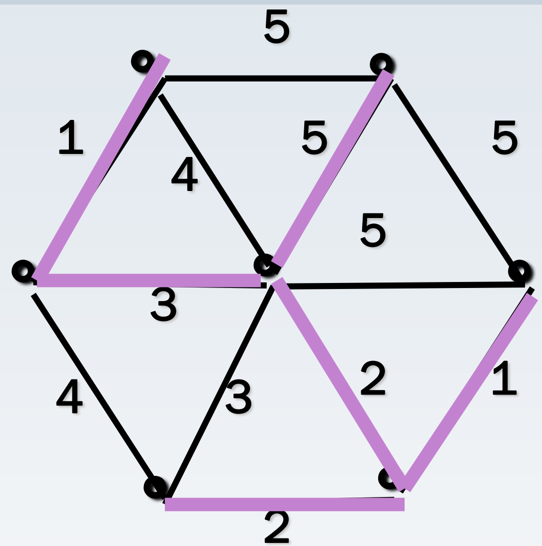


例：用Kruskal算法求下图的最小生成树。



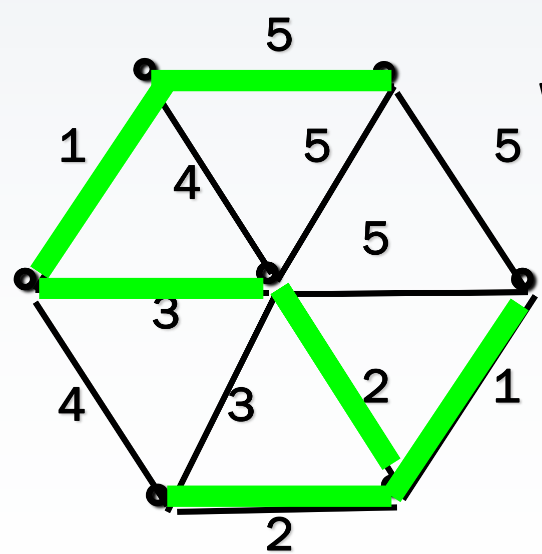
OK!

例：用Kruskal算法求下图的最小生成树。



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

=14



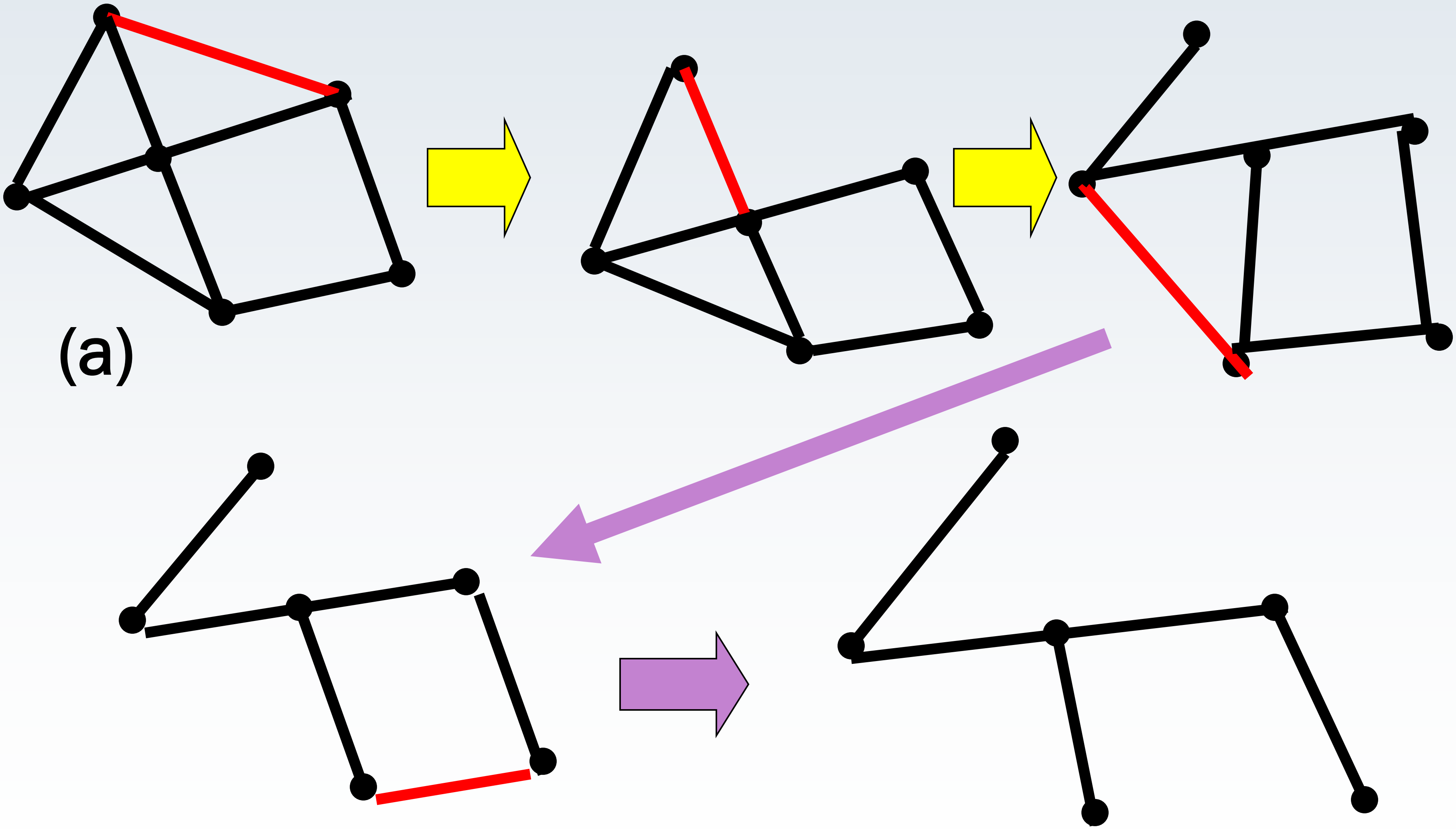
W(T)=1+1+2+3+2+5

=14

破圈法

- 上述算法是贪婪地增加不构成回路的边，以求得最小生成树（最优树），所以通常称为“避圈法”；
- 我们还可以从另一个角度来考虑最小成树（最优树）问题，由G的圈（回路）最优条件，我们也可以在原连通权图G中逐步删除构成回路中权最大的边，最后剩下的无回路的生成子图为最小成树（最优树）。我们把这种方法称为“破圈法”。

例 在图(a)中给出了一个连通图，求此图的生成树



例 用“破圈法”求其最优树的过程。