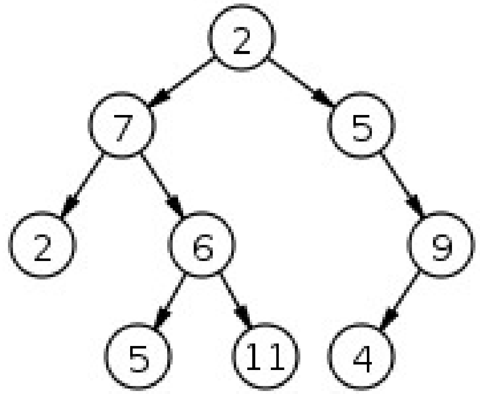
树

树结构简介

- 树结构用图形表示层次结构.
- 称为树(tree)是因为图形的通常画法像一棵树,尽管一般是倒过来的(即树根在上,树叶在下).
- 树结构的例子:
  - 化学:Cayley最早(1857年)用树来研究化合物结构.
  - 计算机科学技术:二叉搜索树,B树,R树,…
  - 信息管理:目录系统
  - 管理:层次式组织机构
  - 生物学:进化树
  - 商务:传销(金字塔销售模式)



树的定义

- 定义:不含回路的(无向)连通图称为树.
  - 树必不含多重边和自环,故是简单图;
  - 边也称为树枝;
  - 度为1的结点称为树叶(*leaf*)或悬挂点;
- 定义:不含回路的图称为森林(*forest*).
  - 森林可能不是连通的.
  - 森林的每个连通支都是树.

割边

- 定义:若$G' = G-e$比G的连通支数多,则称e是G的割边.
  - 删去$e=(u,v)$,则u,v分属于不同的连通支.
- 定理:$e=(u,v)$是割边 *iff* e不属于任何回路.

  $\Rightarrow$:若$e=(u,v)$属于某回路,则$G' = G-e$中仍有从u到v的道路,故u,v属于同一连通支,与e是割边矛盾.

  $\Leftarrow$:若$e=(u,v)$不是割边,则$G' = G-e$中与的连通支数一样,故u,v属于同一连通支,有从u到v的道路,连同e即构成回路,矛盾.
- 显然,树中的边都是割边.

树的等价定义

- 定理:设T是结点数$n\geq 2$的树,则下列性质互相等价:

  (1)T连通且无回路;

  (2)T连通且有$n-1$条边;

  (3)T有$n-1$条边且无回路;

  (4)T连通且每条边都是割边;

  (5)T无回路,但任意增加一边后恰有一个回路;

  (6)T的任意两结点间有唯一道路.

满足连通,无回路,有$n-1$条边之任意两条者是树

树是边数最少的连通图.也是边数最多的无回路图.

树的其他性质

- 设树T的结点数$n \geq 2$,则T中必有树叶.

  证法1:若各结点度都≥2,则总度数$\geq 2n \neq 2m = 2(n-1)$.

  证法2:考虑从任一结点v出发沿边前进,走过的边不重复,则必止步于某树叶.

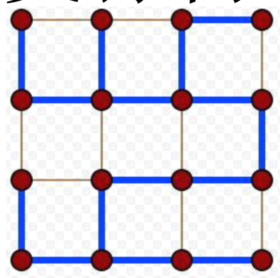
- 设树T的结点数$n \geq 2$,则T至少有两个树叶.

  证明:同上面证法2,考虑从一树叶出发前进,必止步于另一个树叶.

- 若森林F有n个结点和k个连通支,则F有$n-k$条边.

支撑树

- 定义:如果T是图G的支撑子图,而且是树,则称T是G的支撑树或生成树.



- 图G有支撑树 *iff* G是连通的.

  $\Rightarrow$:显然

  $\Leftarrow$:不断对G删除回路即可得到支撑树.

- 若图G本身不是树,则其支撑树不唯一.

二叉树

- 定义:设T是有向树.若T中存在负度为0的结点v_0,其余结点负度为1,则称T是以v_0为根的外向树,或称根树.
- 定义:除树叶外,其余结点的正度最多为2的外向树称为二叉树.
  - 如果它们的正度都是2,称为完全二叉树.

Huffman树

- 赋权二叉树:赋予树叶v_i一个正实数w_i.
  - 从根v_0到树叶v_i的路径$P(v_0,v_i)$的长度l_i:即该路径所含边数.
  - T的加权路径总长度WPL: $\Sigma_i\, l_i w_i$
- 若给定树叶数目及其权值,可以构造许多赋权二叉树,其中必存在WPL最小的二叉树,称为最优二叉树.
  - Huffman算法:给定n个带权树叶,构造最优二叉树(称为Huffman树).

Huffman算法

1.对n个权值排序,得到

$$w_{i_1} \le w_{i_2} \le \ldots \le w_{i_n}.$$

2.计算

$$w_i = w_{i_1} + w_{i_2}$$

作为中间结点v_i的权. v_i的左儿子是 v_{i_1},右儿子是v_{i_2}.

在权序列中删去w_{i_1}和w_{i_2},加入w_i.

$n \leftarrow n-1$,若$n=1$,结束;否则转1.

Huffman算法的正确性

- 定理:由Huffman算法得到的二叉树是最优二叉树.

最短树

- 最短树问题:求赋权连通图的总长最小的支撑树.
  - 相反地:最长树问题.
- 应用:在若干加油站之间铺设输油管道,在确保各加油站供应的条件下,要使建设费用最小.

Kruskal算法

- Kruskal算法:不断加入最短边,并保持无回路.

$T \leftarrow \varnothing$;

WHILE $|T| < n-1 \wedge E \neq \varnothing$ DO

BEGIN

$e \leftarrow E$中最短边;

$E \leftarrow E - e$;

若$T + e$无回路,则$T \leftarrow T + e$;

END;

IF $|T| < n-1$ THEN 打印“非连通”

ELSE 输出最短树T.

Kruskal算法的正确性

- 定理:$T=(V,E')$是赋权连通图$G=(V,E)$的最短树 *iff* 对任意余树边$e\in E-E'$, $E'+e$中的回路C^e满足

$$w(e) \geq w(a), \text{对}\forall a\in C^e, a \neq e.$$

证:$\Rightarrow$若有余树边e满足$w(e)<w(a)$,$\exists a\in C^e$,则以e换a得到的树T'比T更短,矛盾.

$\Leftarrow$对不同于T的支撑树T',有$T'-T\neq\varnothing$, $\forall e\in T'-T$, $T+e$中有回路C^e.据已知条件,对$\forall a\in C^e\cap T$有$w(e)\geq w(a)$,则易证$w(T')\geq w(T)$.

Prim算法

- Prim算法:初始任选一结点,然后不断加入距离最近的结点.

$T \leftarrow \varnothing;\ t \leftarrow v_1;\ U \leftarrow \{t\};$

WHILE $U \neq V$ DO

BEGIN

$w(t,u) = \min_{v \in V-U}\{w(t,v)\};$

$T \leftarrow T + e(t,u);$

$U \leftarrow U + u;$

FOR $v \in V-U$ DO

$w(t,v) \leftarrow \min\{w(t,v), w(u,v)\}$

END;

Prim算法的正确性

- 定理:设V'是赋权连通图G=(V,E)的结点真子集,e是端点分别属于V'和$V-V'$的最短边,则G中一定存在包含e的最短树T.
- 定理:Prim算法可得到赋权连通图G的一棵最短树.

End