离散数学

第八章: 树

卢杨

厦门大学信息学院计算机科学系

luyang@xmu.edu.cn

8.1 无向树

无向树

- 树是一种特殊图，在计算机领域中具有非常重要的应用.
- 本章所讲回路均指初级回路或简单回路.

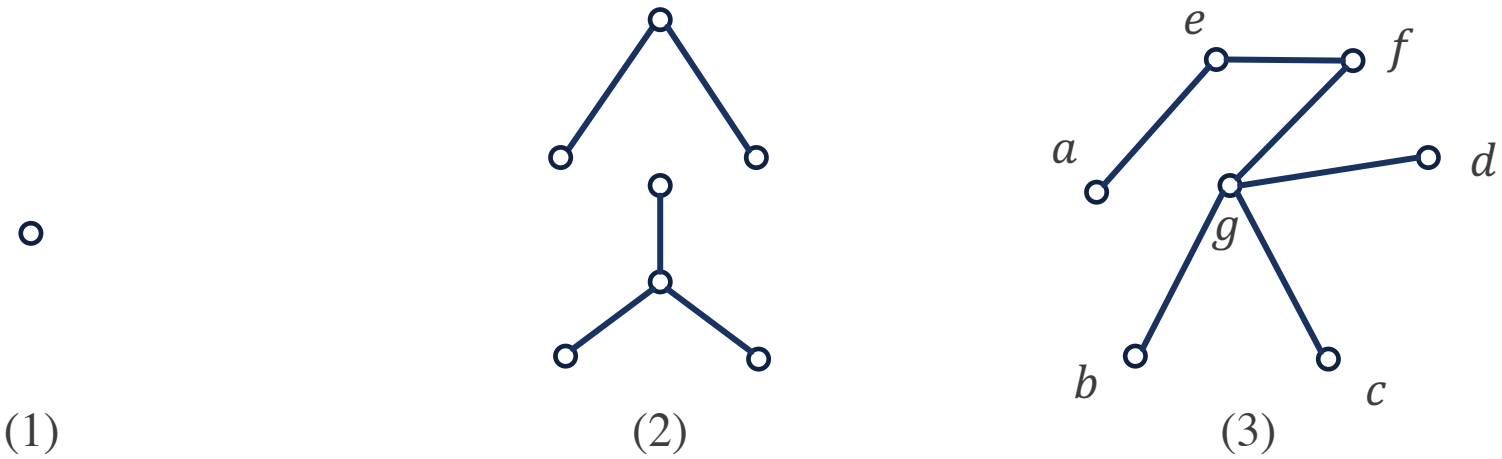
定义 8.1

连通不含回路的无向图称为无向树，简称为树. 常用T表示一颗树. 每个连通分支都是树的非连通无向图称为森林. 平凡图称为平凡树.

- 设$T = \langle V, E \rangle$为一棵无向树，$v \in V$，若$d(v) = 1$，则称v为T的树叶；若$d(v) \geq 2$，则称v为T的分支点.

无向树

- (1)是平凡树; (2)是2棵树组成的森林; (3)是1棵无向树.
- (3)中, a, b, c, d均为树叶; e, f, g均为分支点.



(1) (2) (3)

无向树

定理 8.1

设$G = \langle V, E \rangle$, $|V| = n$, $|E| = m$下列各命题等价.

(1) G是树 (G连通且不含回路);

(2) G的每对顶点之间存在唯一的一条初级路径;

(3) G是连通的, 且$m = n - 1$;

(4) G中无回路, 且$m = n - 1$;

(5) G中无回路, 但在G中任何两个不相邻的顶点之间增加一条新边, 就得到唯一的一条初级回路;

(6) G是连通的, 且G中任何边均是桥. (树T中分支点必为割点. 树是最弱的连通图).

- 我们使用循环证明法来证明该定理.

无向树

(1) G是树 (G连通且不含回路);

(2) G的每对顶点之间存在唯一的一条初级通路;

证明 (1)⇒(2)

- 首先证明G的每对顶点之间存在初级通路. 设u, v为G中任意两个顶点. 由于G是连通的, 所以u, v之间有通路. 若该通路不是初级的, 则必然存在某个v'在该通路中出现2次, 则G中必有回路, 与G不含回路矛盾, 因此该通路是初级通路.
- 再证明该初级通路是唯一的. 若u, v之间的初级通路多于一条, 必形成回路, 这与G中不含回路矛盾, 所以G的每对顶点之间存在唯一的一条初级通路.

无向树

(2) G的每对顶点之间存在唯一的一条初级通路;

(3) G是连通的, 且$m = n - 1$;

证明 (2)⇒(3)

- 首先证明G是连通的. 由于G中任意两个顶点之间均有初级通路, 所以任意两个顶点均是连通的, 所以G是连通的.
- 再通过归纳法证明$m = n - 1$.
  - 当$n = 1$时, G为平凡树, $m = 0$, 结论显然成立.
  - 假设当$n \leq k$时结论成立, 证明$n = k + 1$时结论也成立.
    - 设$e = (u, v)$为G中的一条边, 由(2)可知uv是唯一的初级通路, 因此$G - e$有两个连通分支G_1与G_2.
    - 设它们的顶点数为n_1, n_2, 边数为m_1, m_2, 显然$n_1 \leq k, n_2 \leq k$, 且G_1和G_2都有性质(2).
    - 由归纳假设得$m_1 = n_1 - 1, m_2 = n_2 - 1$. 从而$m = m_1 + m_2 + 1 = n_1 - 1 + n_2 - 1 + 1 = n - 1$.

无向树

(3) G是连通的, 且$m = n - 1$;

(4) G中无回路, 且$m = n - 1$;

证明 (3)⇒(4)

只需要证明G中无回路.

- 若G中有回路, 从回路中删去任意一条边后, 所得图仍然连通, 若所得图中再有回路, 再从回路中删去一条边, 直到所得图中无回路为止.
- 设共删去$r(r \geq 1)$条边所得图为G'. G'无回路, 但仍是连通的, 即G'是树.
- 由(1)⇒(2)⇒(3), 可知G'中$m' = n' - 1$. 而$n' = n, m' = m - r$. 于是得到$m - r = n - 1$, 即$m = n - 1 + r\ (r \geq 1)$, 这与已知条件矛盾.

无向树

(4) G中无回路, 且$m = n - 1$;

(5) G中无回路, 但在G中任何两个不相邻的顶点之间增加一条新边, 就得到唯一的一条初级回路;

证明 (4)⇒(5)

- 首先证明G是连通的.
  - 假设G是非连通的, 则G有$k(k \geq 2)$个连通分支$G_1, G_2, \ldots, G_k$. 设G_i有n_i个顶点, m_i条边.
  - 由于G中无回路, 故连通分支G_i中也无回路, 因此每个连通分支G_i都是树.
  - 由(1)⇒(2)⇒(3), 可得$m_i = n_i - 1$, 于是$n = n_1 + n_2 + \cdots + n_k = m_1 + 1 + m_2 + 1 + \cdots + m_k + 1 = m + k\ (k \geq 2)$. 这与已知$m = n - 1$矛盾, 因此$G$是连通的.
- 因而G是连通的, 又是无回路的, 即G是树. 由(1)⇒(2), G中任意两个不相邻的顶点u, v之间存在唯一的初级通路, 在u与v之间加上边就得到唯一的一条初级回路.

无向树

(5) G中无回路, 但在G中任何两个不相邻的顶点之间增加一条新边, 就得到唯一的一条初级回路;

(6) G是连通的, 且G中任何边均是桥. (树T中分支点必为割点. 树是最弱的连通图).

证明 (5)⇒(6)

- 首先证明G是连通的.
  - 假设G是非连通的, 设G_1, G_2是其中的两个连通分支. v_1为G_1中的一个顶点, v_2为G_2中的一个顶点.
  - v_1与v_2不相邻, 但是在G中加边(v_1, v_2)不形成回路, 这与已知条件矛盾, 所以G是连通的.
- 然后证明G中任何边均是桥.
  - 若G中存在一条边$e = (u, v)$不是桥, 即$G - e$仍连通, 说明在$G - e$中存在u到v的通路.
  - 此通路与e构成G中的回路, 这与G中无回路矛盾.

无向树

(6) G是连通的, 且G中任何边均是桥. (树T中分支点必为割点. 树是最弱的连通图).

(1) G是树 (G连通且不含回路);

证明 (6)⇒(1)

只需证明G中无回路.

- 若G中含回路C, 删除C上任何一条边后, 所得的图仍连通, 这与G中任何边均是桥矛盾, 因此G中不含回路, 即G是树.

无向树

定理8.2

设T是n阶非平凡的无向树, 则T至少有两片树叶.

证明 在非平凡树中, 任何顶点的度数均大于等于1.

设G中有k个1度顶点, 即k片树叶, 则其余$n-k$个分支点的度数均大于等于2.

由握手定理可知

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

由定理8.1知$m = n-1$, 代入上式可得

$$2n - 2 \geq k + 2n - 2k,$$

解得$k \geq 2$. 说明T至少有两片树叶.

无向树

推论 阶大于2的树T必有割点.

证明

由$m = n - 1$可知, n永远比m多1, 根据握手定理$\sum_{i=1}^{n} d(v_i) = 2m = 2(n-1)$, 在$n > 2$时, 度数和$2(n-1) > n$, 即$T$至少含有一个度数≥2的分支点$v$.

再由定理8.1 (T是连通的, 且T中任何边均是桥), 可得$T - \{v\}$不是连通的, 所以v是割点. (如果v是悬挂点,$T - \{v\}$还是连通的, 所以要先证明$d(v) \geq 2$)

推论 树中分支点必为割点, 树中边均为桥.

- 树是连通性最弱的连通图($n = m - 1$), 但也是最经济的连通图.
  - 在保证连通性的前提下, 边没办法更少了.

无向图

例 8.1 已知一棵无向树T中4度, 3度, 2度的分支点各一个, 其余的顶点均为树叶, 问T中有几片树叶?

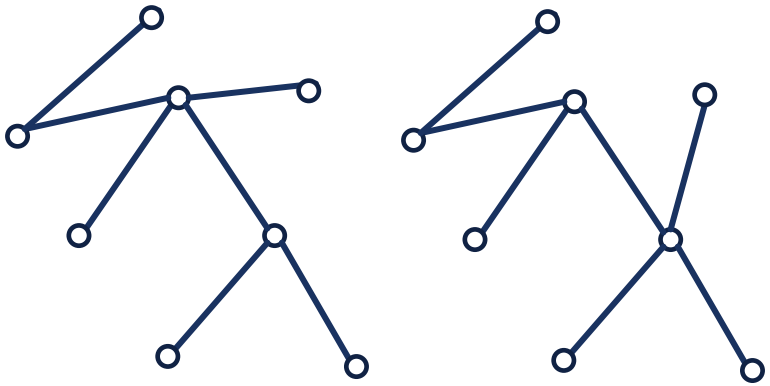
解 设T中有x片树叶. 则T的阶数$n = 1 + 1 + 1 + x = 3 + x$. 由$m = n - 1$可得$m = 2 + x$. 每个树叶都是1度, 由握手定理有总度数

$$4 + 2x = 2m = 4 + 3 + 2 + x = 9 + x,$$

解得$x = 5$片树叶.

例 8.2 满足例8.1中度数列(4, 3, 2, 1, 1, 1, 1, 1)的无向树在同构的意义下是惟一吗?

解 在同构意义下不惟一, 如图所示.



生成树

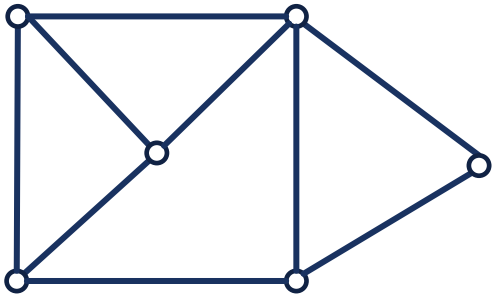
定义 8.2

设$G = \langle V, E \rangle$是无向连通图, T是G的生成子图并且为树, 则称T是G的生成树. 对任意的边$e \in E(G)$, 若$e \in E(T)$, 则称e为T的树枝, 否则称e为生成树T的弦. 称所有弦的集合导出子图$G[E(G) - E(T)]$为生成树T的余树, 记作$\overline{T}$.

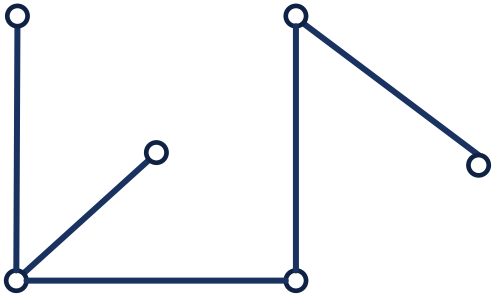
- 余树不一定连通, 也不一定不含回路, 因此不一定是树.
- 一般来说, 连通图的生成树不是唯一的.
- 一个图G与它的生成树的差别在于G可能包含有回路, 而生成树不包含有回路.

生成树

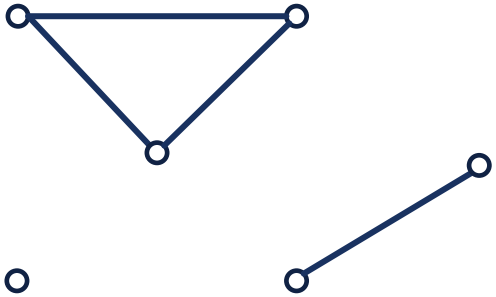
例 下图中, (2)是(1)的一颗生成树, (3)是(1)的余树.



(1)



(2)



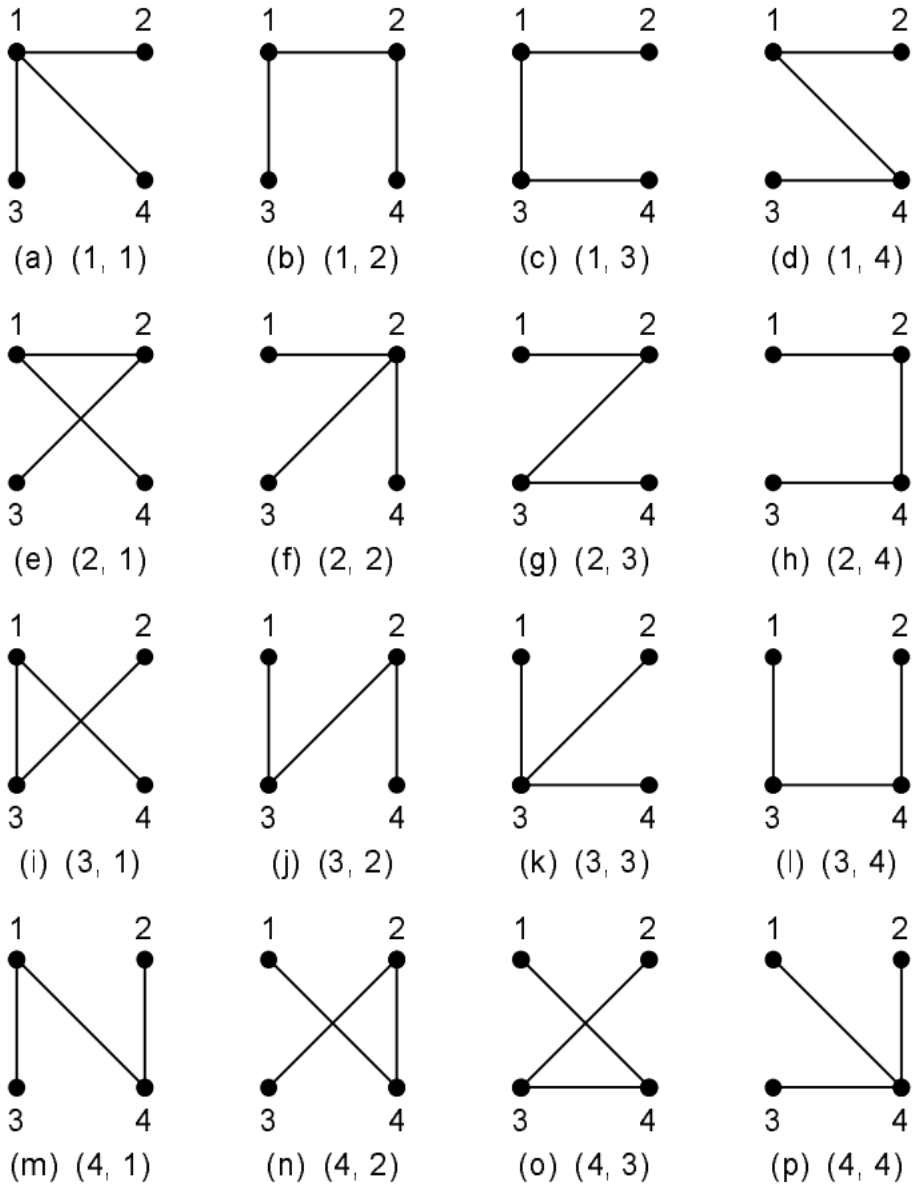
(3)

课堂练习

画出K_4的所有生成树. 其中非同构的有几种?

解

16个生成树, 其中2种非同构



Source: https://www.csie.ntu.edu.tw/~kmchao/tree07spr/counting.pdf

生成树

定理 凯莱公式

K_n的生成树的个数是n^{n-2}.

定理 8.3

任何无向连通图G都存在生成树.

证明

- 若G中无回路, 由于G是连通的, 所以G是树, 所以G就是自己的生成树.
- 若G中有回路C, 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, 就继续在回路中删除一条边直到没有回路. 设最后的图为G', G'是连通的且无回路, 所以G'是G的生成树.

基本回路

定理

设T是无向连通图G中的一棵生成树, e为T的任意一条弦, 则$T \cup e$中含G的只含一条弦其余边均为树枝的初级回路, 而且不同的弦对应的初级回路是不同的.

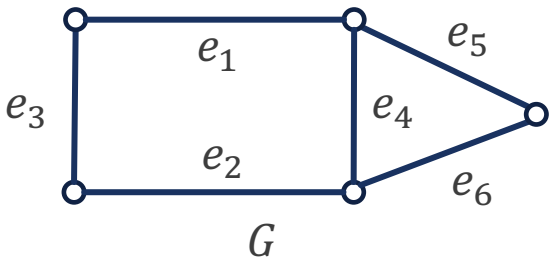
证明 设$e = (u, v)$为T中的任意一条弦.

- 由于e是弦, 所以u和v在T中不相邻.
- 由定理8.1(6)可知, u与v之间增加一条新边, 也就是e, 放入T中后得到一条唯一的初级回路. 该回路中只含一条弦, 其余边均为树枝.
- 当e_1, e_2为任意不同的弦时
  - e_2不在e_1对应的圈C_{e_1}中, 因为圈C_{e_1}中只含一条弦, 就是e_1;
  - e_1不在e_2对应的圈C_{e_2}中, 因为圈C_{e_2}中只含一条弦, 就是e_2.

基本回路

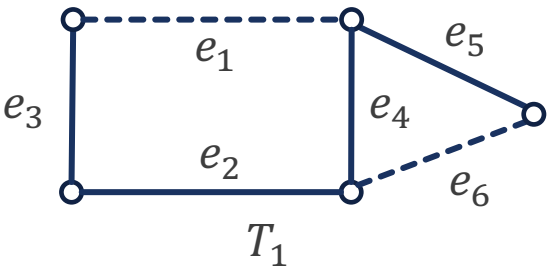
定义 8.3

设G是m条边的n阶无向连通图, T是G的一棵生成树, T的$m-n+1$条弦为$e_1, e_2, \dots, e_{m-n+1}$. G中仅含T的一条弦e_r的回路C_r称作对应弦e_r的基本回路, $\{C_1, C_2, \dots, C_{m-n+1}\}$称作$G$对应$T$的基本回路系统.



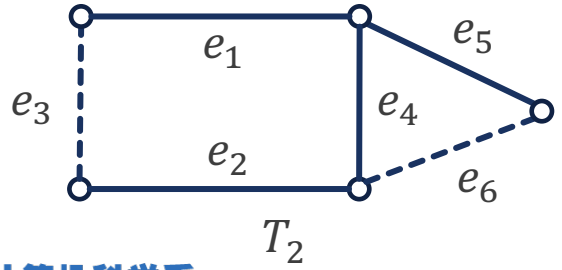
G

例 8.3 求该图G的两个生成树T_1, T_2的基本回路系统.



T_1

解 T_1: $\{e_1e_4e_2e_3, e_6e_4e_5\}$; T_2: $\{e_3e_1e_4e_2, e_6e_4e_5\}$.



T_2

- 基本回路的个数和弦数是相同的, 都等于$m-n+1$.

基本割集

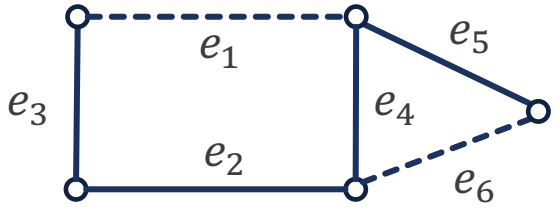
- 设T是连通图G的一棵生成树，e是一条树枝. 根据定理 8.1(6)，$T-e$是不连通的，它有两个连通分支T_1和T_2. 于是，由e和T的弦就可以构成一个G的割集.

定义 8.4

设T是n阶连通图G的一棵生成树，$e'_1, e'_2, \ldots, e'_{n-1}$为$T$的树枝. 设$S_r$是只含树枝$e'_r$其余边均为弦的$G$的割集，则称$S_r$为$G$的对应树枝$e'_r$的基本割集，并称$\{S_1, S_2, \ldots, S_{n-1}\}$为$G$对应$T$的基本割集系统.

- 基本割集的个数和树枝数是相同的，都等于$n-1$.

例 在该图中，对应e_5, e_4, e_2, e_3的基本割集系统为$\{\{e_5, e_6\}, \{e_4, e_1, e_6\}, \{e_2, e_1\}, \{e_3, e_1\}\}$.



基本割集

求树枝e对应的基本割集的方法如下:

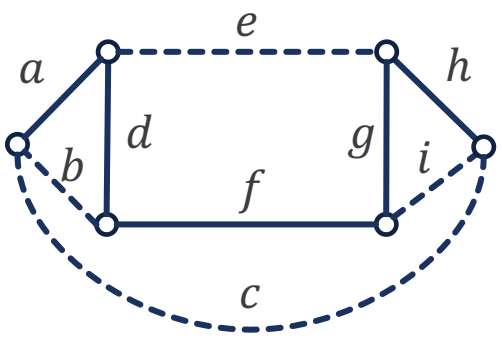
- 将e从T中删除,得到$T-e$的两个连通分支为T_1, T_2,设T_1和T_2的顶点集为V_1, V_2.
- 则e对应的基本割集为

$$S_e = \{e' \mid e' \in E(G) \wedge$$
$$e\text{的一个端点在}V_1\text{中,另一个端点在}V_2\text{中}\}.$$

基本回路与基本割集

例 8.4 求该图G的生成树T的基本回路系统和基本割集系统.

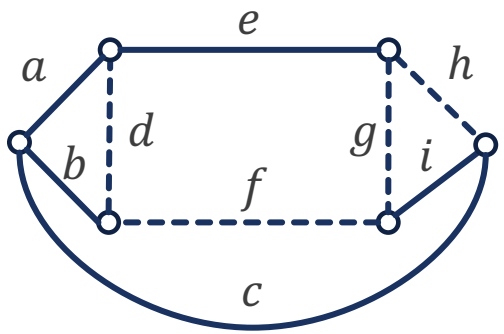
解 首先确定这两个系统中元素的个数. G的顶点数$n=6$, 边数$m=9$, 基本回路个数为$m-n+1=4$, 基本割集个数为$n-1=5$.



- T的4条弦e,b,c,i对应的基本回路为$edfg, bda, cadfgh, igh$.
- T的5条树枝a,d,f,g,h对应的基本割集为$\{a,b,c\},\{d,e,b,c\},\{f,e,c\},\{g,e,i,c\},\{h,i,c\}$.

课堂练习

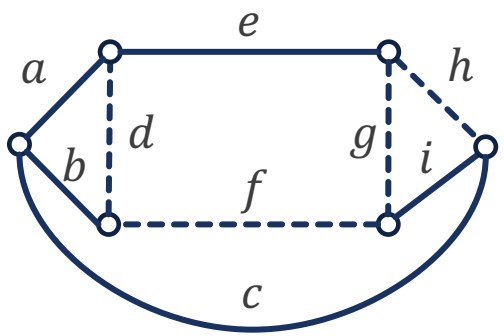
求该图G的生成树T的基本回路系统和基本割集系统.



课堂练习

求该图G的生成树T的基本回路系统和基本割集系统.

解 首先确定这两个系统中元素的个数. G的顶点数$n=6$, 边数$m=9$, 基本回路个数为$m-n+1=4$, 基本割集个数为$n-1=5$.



- T的4条弦d, f, g, h对应的基本回路为$dab, fbci, gicae, hcae$.
- T的5条树枝a, e, b, c, i对应的基本割集为$\{a,d,g,h\}, \{e,g,h\}, \{b,d,f\}, \{c,f,g,h\}, \{i,g,f\}$.

最小生成树

- 想给城镇之间铺路，把若干城市连接起来（成为连通图），怎样才能使所用的线路总长最短（或时间最少，资源最少）呢？问题的实质就是求带权的最小生成树问题.

定义 8.5

对图G的每条边e附加一个实数$w(e)$，称$w(e)$为边e的权. G连同附加在各边上的权称为带权图，常记作$G = \langle V, E, W \rangle$. 设$G_1 \subseteq G$，称$\sum_{e \in E(G_1)} w(e)$为$G_1$的权，记作$W(G_1)$.

定义 8.6

设无向连通带权图$G = \langle V, E, W \rangle$，$G$的所有生成树中$W(T)$最小的生成树$T$称为$G$的最小生成树.

避圈法

- 设n阶简单无向连同带权图$G = \langle V, E, W \rangle$有$m$条边. 下面介绍求最小生成树的算法: 避圈法 (Kruskal算法).

  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. 令$T = \emptyset$.

  3. 依次取边并判断$T \cup e_i$是否有回路.

  4. 若无回路, 则更新$T \leftarrow T \cup e_i$; 若有回路, 则弃去e_i.

- 那么, 为何该算法总能够得到最小生成树? 该算法是否是最高效的呢?

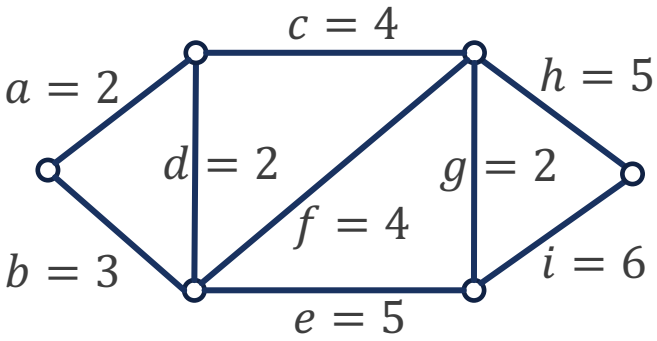
避圈法

- 对于计算机，判断加入新的边之后T是否有回路的简单方法是判断该边的两个端点是否在同一集合内.

```
void kruskal (int n, int m,
              edge_set E,
              edge_set& F)
{
    index i, j;
    set_pointer p, q;
    edge e;
    sort the m edges in E by weight in nondecreasing order;
    F = ø;
    initial(n);
    while (number of edges in F is less than n - 1){
        e = edge with least weight not yet consisdered;
        i, j = indices of vertices connected by e;
        p = find(i);
        q = find(j);
        if (!equal(p, q)){
            merge(p, q);
            add e to F;
        }
    }
}
```

避圈法

例 8.5 通过避圈法求该图的最小生成树.

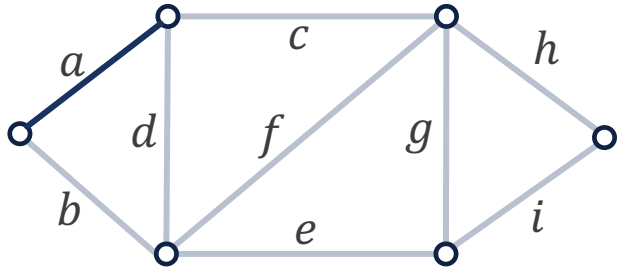


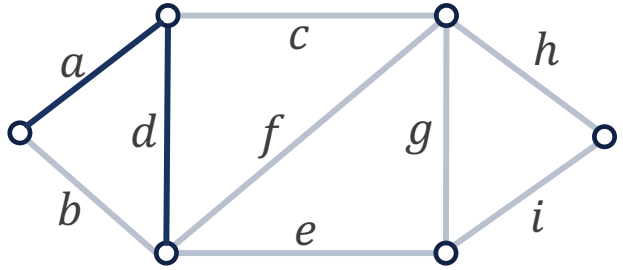
解

先对边的权进行排序:
$a, d, g, b, c, f, e, h, i$.

(1) 加入$a, T = \{a\}$.

(2) 加入$d, T = \{a, d\}$.





避圈法

序列: $a, d, g, b, c, f, e, h, i$.

(3) 加入g, $T = \{a, d, g\}$.

(4) 加入b, 有回路, 扔掉.

(5) 加入c, $T = \{a, d, g, c\}$.

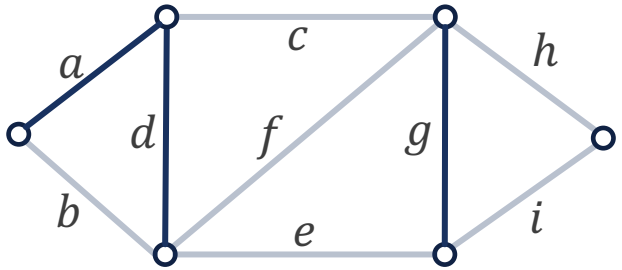
(6) 加入f, 有回路, 扔掉.

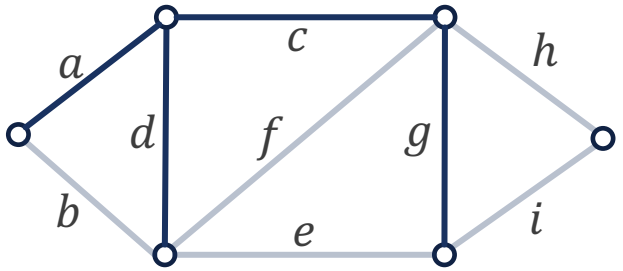
(7) 加入e, 有回路, 扔掉.

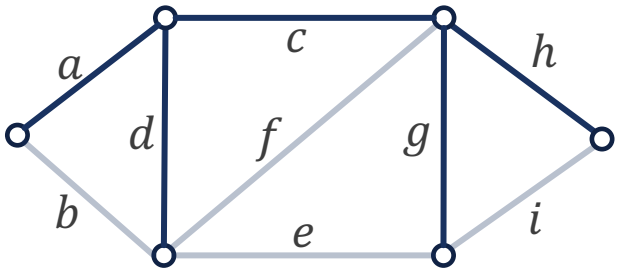
(8) 加入h, $T = \{a, d, g, c, h\}$.

(9) 加入i, 有回路, 扔掉.

最终该生成树的权为15, 是最小的.

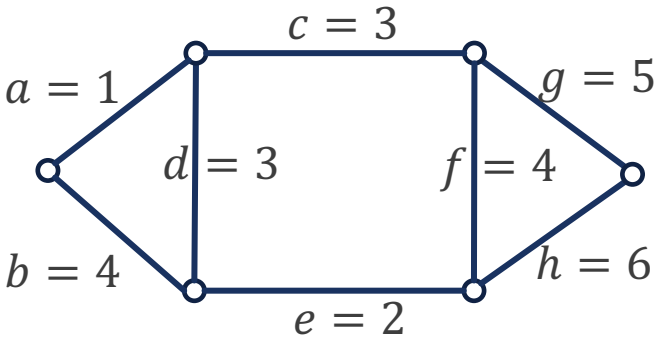






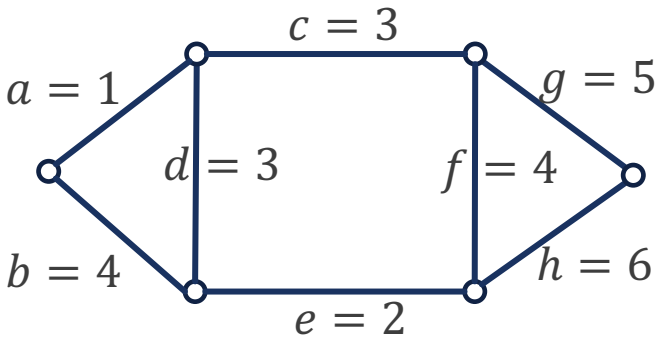
课堂练习

通过避圈法求该图的最小生成树.



课堂练习

通过避圈法求该图的最小生成树.

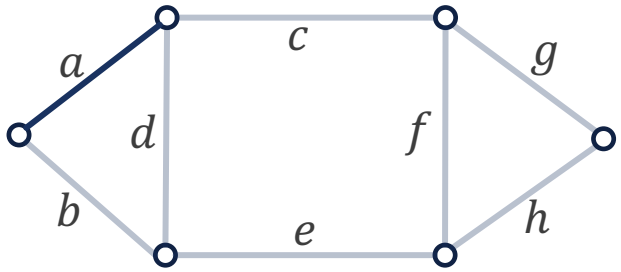


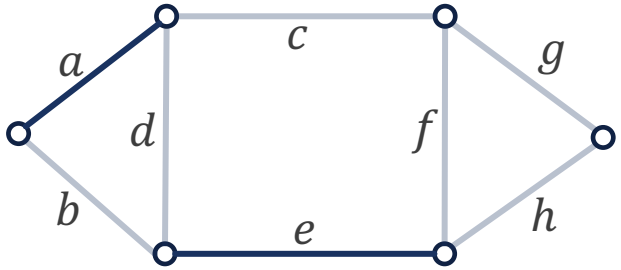
解

先对边的权进行排序:
a, e, c, d, b, f, g, h.

(1) 加入$a, T=\{a\}$.

(2) 加入$e, T=\{a, e\}$.





课堂练习

序列: a, e, c, d, b, f, g, h.

(3) 加入c, $T = \{a, e, c\}$.

(4) 加入d, $T = \{a, e, c, d\}$.

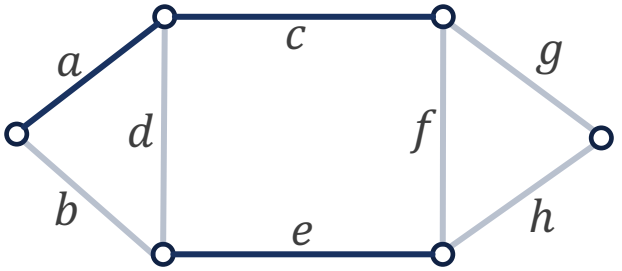
(5) 加入b, 有回路, 扔掉.

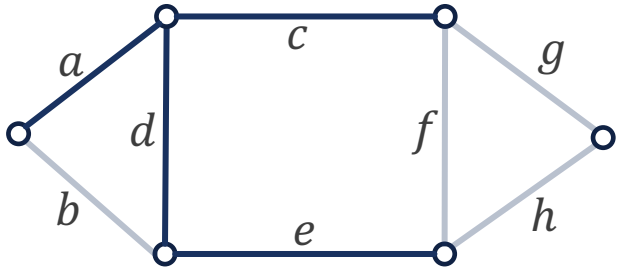
(6) 加入f, 有回路, 扔掉.

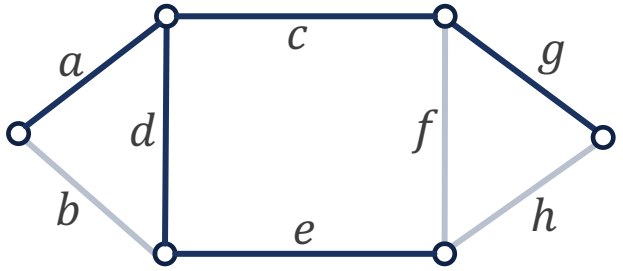
(7) 加入g, $T = \{a, e, c, d, g\}$.

(8) 加入h, 有回路, 扔掉.

最终该生成树的权为14, 是最小的.







8.2 根树及其应用

根树

- 若有向图D的基图是无向树，则称D为有向树. 在有向树中，最重要的是根树，它在数据架构，算法设计，数据库等专业课程中极其重要.

定义 8.7

设有向图树T是一颗非平凡树，如果T中有一个顶点的入度为0，其余顶点的入度均为1，则称T为根树.

在根树中，入度为0的顶点称为树根；入度为1且出度为0的顶点称为树叶；入度为1且出度大于0的顶点称为内点. 内点和树根统称为分支点.

在根树中，从树根到任一顶点v的通路长度称为v的层数，记作$l(v)$；称层数相同的顶点在同一层上，层数最大顶点的层数为树高，记作$h(T)$.

根树

- 在根树中，由于各有向边的方向的一致性，因而画根树时，省掉有向边的箭头，并将树根放在最上方，边的方向一律向下或斜下方.

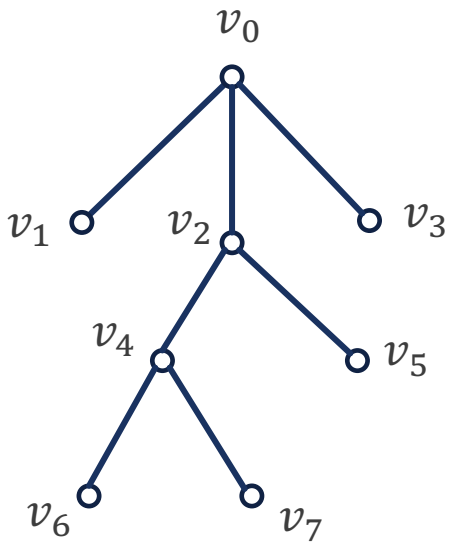
例 该图为一颗根树.

v_0为树根，v_1, v_3, v_5, v_6, v_7均为树叶，v_2, v_4为内点，v_0, v_2, v_4为分支点.

各顶点层数:

- $l(v_0) = 0$;
- $l(v_1) = l(v_2) = l(v_3) = 1$;
- $l(v_4) = l(v_5) = 2$;
- $l(v_6) = l(v_7) = 3$.

树高为3，即$h(T) = 3$.



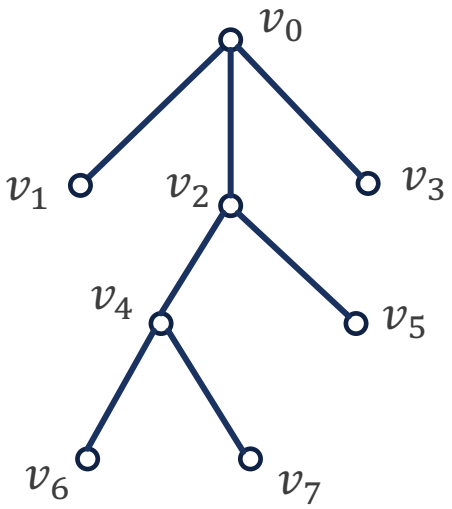
根树

定义

若顶点a邻接到顶点b，则称b为a的儿子，a为b的父亲；

若b, c的父亲相同，则称b, c为兄弟；

若$a \neq d$，而a可达d，则称a为d的祖先，d为a的后代.



例 该树中，v_1, v_2, v_3是兄弟，它们的父亲是v_0；

v_4和v_5是兄弟，它们的父亲是v_2；

v_6和v_7是兄弟，它们的父亲是v_4；

v_0以外的所有顶点都是v_0的后代，v_0是它们的祖先.

有序树, 正则树, 完全树

定义 8.8

如果将根树T的每一层上的顶点都规定次序, 则称T为有序树.

定义 8.9

设T为一棵非平凡的根树,

(1) 若T的每个分支点至多有r个儿子, 则称T为r元树;

(2) 若T的每个分支点都恰有r个儿子, 称T为r元正则树;

(3) 若r元树T是有序的, 则称T为r元有序树;

(4) 若T是r元正则树, 且所有树叶的层数均为树高, 则称T为r元完全正则树;

(5) 若T是r元完全正则树, 且T是有序的, 则称T为r元完全正则有序树.

- 完全树一定是正则树, 但是正则树不一定是完全的.

2叉树

- 在所有的r元树中，2元树最重要，2元树又称2叉树.

定义

对一棵根树的每个顶点都访问一次且仅访问一次称为遍历.

定义

设T为一棵根树，a是T的一个非根顶点，称a及其后代的导出子图T'为T的以a为根的根子树.

- 2元正则有序树的每个分支点的两个儿子导出的根子树分别称为该分支点的左子树和右子树.

2叉树的遍历

- 设T为一棵2元正则有序树, 按对树根, 左子树, 右子树的不同的访问顺序主要有以下3种遍历方法:

(1) 中序遍历法: 访问次序为左子树, 树根, 右子树;

(2) 前序遍历法: 访问次序为树根, 左子树, 右子树;

(3) 后序遍历法: 访问次序为左子树, 右子树, 树根.

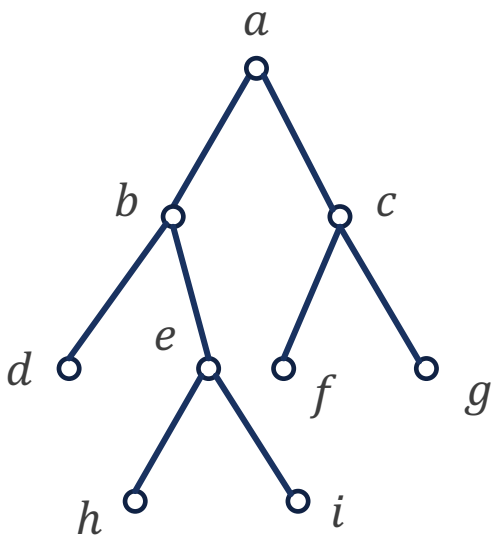
- 这里的前中后指的是树根在访问次序中的位置.

例 该根树按中序, 前序, 后序遍历的结果分别为:

$$(db(hei))a(fcg),$$

$$a(bd(ehi))(cfg),$$

$$(d(hie)b)(fgc)a.$$

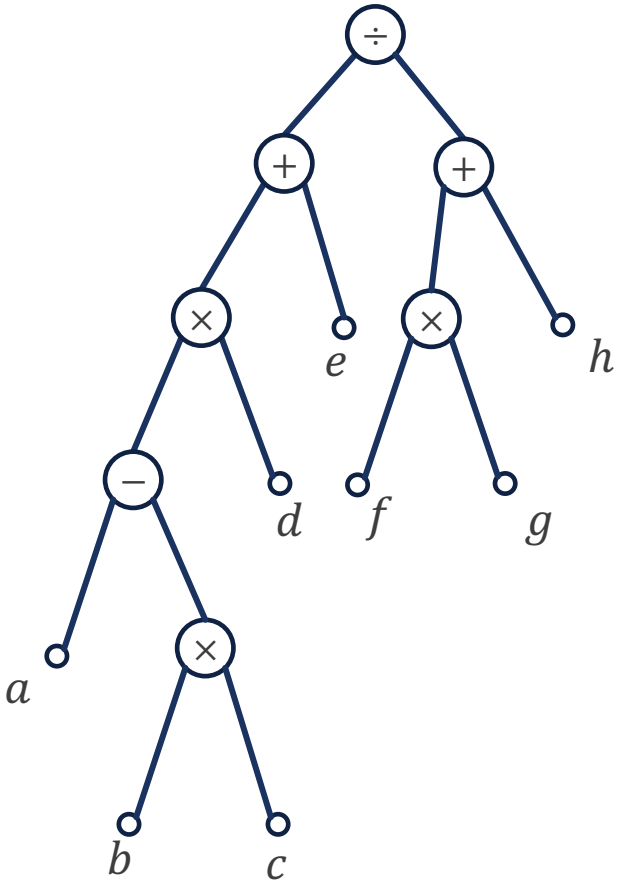


2叉树的遍历

例 8.6(1) 利用2元有序树表示下面算式:

$$\big((a-b\times c)\times d+e\big)\div(f\times g+h).$$

解 将运算符都作为分支点, 运算的数都作为树叶, 可以表示为该二元有序树.



2叉树的遍历

例 8.6(2) 用3种遍历法访问该2元有序树，写出访问结果.

解

- 中序遍历:

$$\Big(\big((a-(b\times c))\times d\big)+e\Big)\div\big((f\times g)+h\big)$$

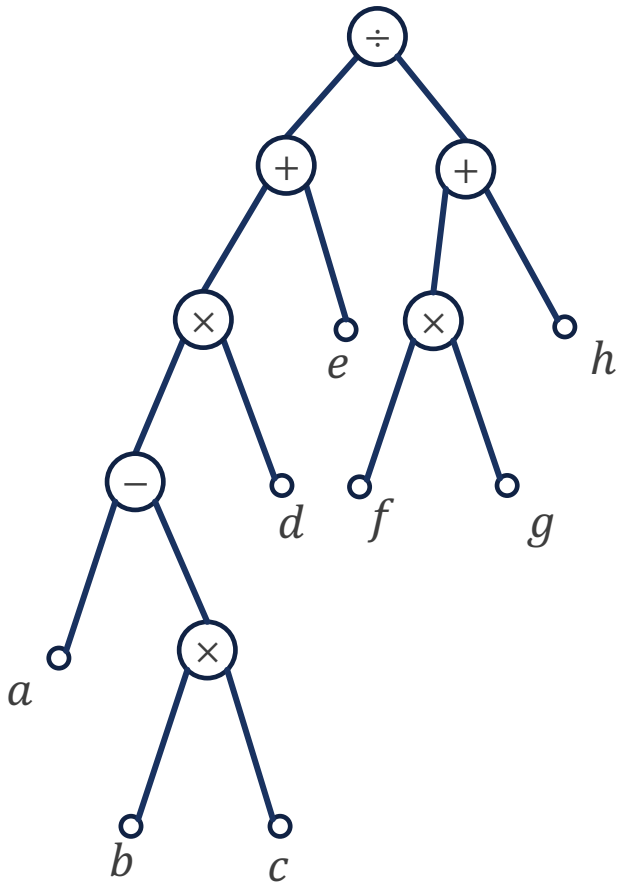
- 前序遍历:

$$\div\big(+\big(\times\big(-a(\times bc)\big)d\big)e\big)(+(\times fg)h)$$

- 后序遍历:

$$\Big(\big((a(bc\times)-)d\times\big)e+\Big)\big((fg\times)h+\big)\div$$

- 中序遍历的结果去掉一些多余的括号后原式结果相同，所以中序遍历访问的结果是还原算式.



2叉树的遍历

- 对于前序遍历的访问结果, 可将全部括号去掉, 得到:

$$\div + \times - a \times bcde + \times fgh.$$

对这个表达式规定, 从右到左, 每个运算符与它后面紧邻的2个数进行运算. 这种运算符在参加运算的数的前面的表达方式称为前缀符号法, 又称作波兰符号法.

- 对于后序遍历的访问结果, 可将全部括号去掉, 得到:

$$abc \times - d \times e + fg \times h + \div .$$

对这个表达式规定, 从左到右, 每个运算符与它前面紧邻的2个数进行运算. 这种运算符在参加运算的数的后面的表达方式称为后缀符号法, 又称作逆波兰符号法.

课堂练习

给定该算式

$$\Big(\big(a\times(b+c)\big)\times d-e\Big)\div(f+g)\div\big(h\times(i+j)\big)$$

(1) 将以上算式存入一棵2元正则有序树中;

(2) 分别写出上式的波兰符号法和逆波兰符号法表达的形式.

课堂练习

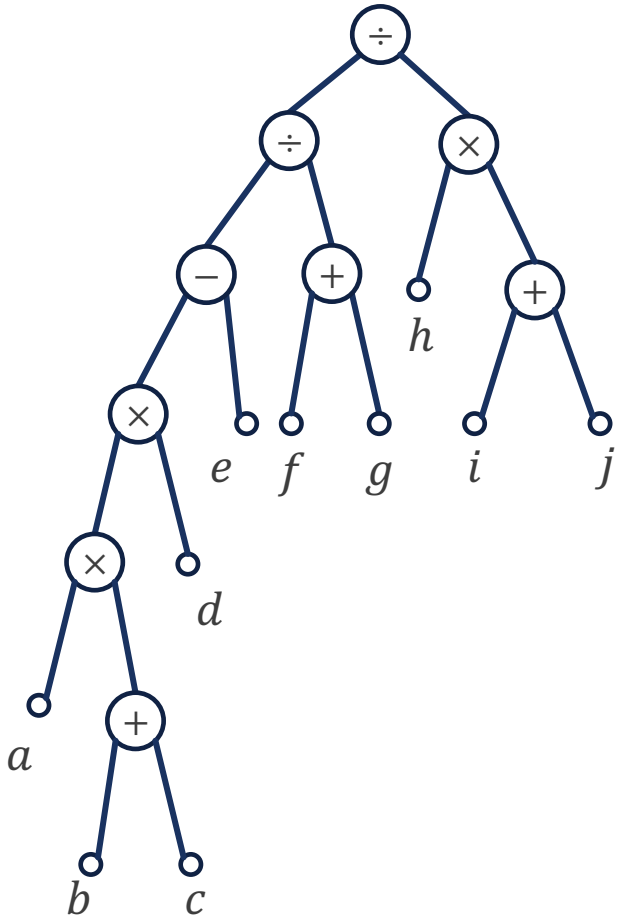
解

波兰符号法表达的形式为

$$\div\div-\times\times a+bcde+fg\times h+ij$$

逆波兰符号法表达的形式为

$$abc+\times d\times e-fg+\div hij+\times\div$$



作业

p172

- 2(2)
- 3
- 4
- 5
- 6
- 8
- 10

谢谢

有问题欢迎随时跟我讨论