

图与组合数学（图论部分）

西安电子科技大学数学系

参考文献

- [1] J.A. Bondy, U.S.R. Murty, Graph theory, Springer, 2008 (**GTM 244**).
- [2] T.H. Cormen et al, Introduction to algorithms (2nd edition.), The MIT Press, 2001.
(中译本: 潘金贵等译, 算法导论, 北京, 机械工业出版社, 2006)
- [3] D. Jungnickel, Graphs, networks and algorithms(3rd edition), Springer, 2008.
- [4] F.S. Roberts, B.Tesman, Applied combinatorics(2nd edition.), Pearson Education, 2005.
(中译本: 冯速译, 应用组合数学, 北京, 机械工业出版社, 2007)
- [5] D.B. West, Introduction to graph theory (2nd edition.), Pearson Education, 2001.
(中译本: 李建中, 骆吉洲译. 图论导引, 北京, 机械工业出版社, 2006)

目 录

第一章 图的基本概念	1
1.1 图与有向图	1
1.2 树的性质	5
1.3 根树及其应用	6
1.4 最小生成树及最短路径问题	6
第二章 几类特殊的图	9
2.1 二部图	9
2.2 哈密尔顿(Hamilton)图与欧拉(Eular)图	16
2.3 彼得森(Petersen)图	16
第三章 图的性质	23
3.1 图的连通	23
3.2 图的着色	24
3.3 可平面图	26
第四章 鸽笼原理与容斥原理	31
4.1 鸽笼原理	31
4.2 容斥原理	35

第一章 图的基本概念

1.1 图与有向图

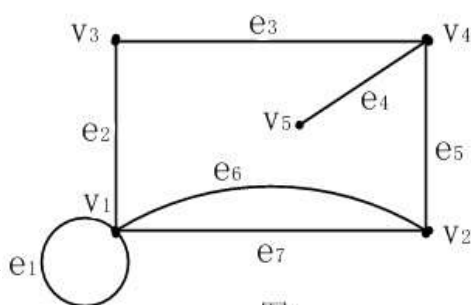
定义 1.1.1 一个无向图，常简称图 (graph)， G 是指一个二元组 (V, E) ，其中集合 V 中的元素称为顶点 (或点, 或端点, 或结点) (or vertex, or node, or point)，集合 E 中元素为 V 中元素组成的无序对，称为边 (edge)。

注记 1.1.2 1. 上述集合 E 中的元素可以相同，有的文献称这样的集合为多重集 (multiset)。

2. 图 (\emptyset, \emptyset) 称为空图，它有时在举反例的时候用到。有时将一个结论推广到包含空图时会引起不必要的麻烦，本讲义中假设所讨论的图都不是空图。

3. 在一个图 $G = (V, E)$ 中，为了表示 V 和 E 分别是 G 顶点集和边集，常将 V 和 E 分别记为 $V(G)$ 和 $E(G)$ 。如果 $V(G)$ 和 $E(G)$ 都是非空有限集，则称 G 为有限图。我们约定除非特别指出，本书中每一个图都是有限图。

我们经常用图形来表示一个图。用小圆圈或实心点表示图的顶点，用线段把无序对中两个顶点连接起来表示边。其中顶点的位置，连线的曲直、是否相交等都无关紧要。例如， $G = (V, E)$ ， $V = \{v_1, v_2, v_3, v_4, v_5\}$ ， $E = \{(v_1, v_1), (v_1, v_2), (v_1, v_2), (v_1, v_3), (v_2, v_4), (v_3, v_4), (v_4, v_5)\}$ ，则 G 的图形如下：



设 $G = (V, E)$ 。若 V 为 V (vertex set) 的集合， E 为 E (edge set) 的集合。为方便起见，

图1

常用 e_i 表示边，例如在图 1 中 e_2 表示边 (v_1, v_3) ，而 v_1, v_3 称为 e_2 的端点。两个端点相同的边称为环 (loop)，具有相同端点的多条边称为重边 (multiple edge)，不含环和重边的图称为简单图 (simple graph)。例如在图 1 中 $e_1 = (v_1, v_1)$ 为环， e_6, e_7 为重边，所以此图不是简单图。一条边的两个顶点称为相邻的 (adjacent)。若 v 是边 e 的一个顶点，则称 v 和 e 是关联的 (incident)。对于无环图，和 v 关联的边的个数称为 v 的度 (degree)，记为 $d(v)$ 。对于有环图，环记 2 度。在图 1 中， $d(v_1) = 5$ ，

$$d(v_2) = 3$$

关于顶点度数，我们有下面的基本定理.

定理 1.1.3 设图 G 的顶点集为 $V(G) = \{v_1, v_2, \dots, v_n\}$ ，且 $|E(G)| = m$ ，则

$$\sum_{i=1}^n d(v_i) = 2m.$$

推论 1.1.4 图中度为奇数的顶点个数为偶数.

我们称一个图的所有顶点度数的非递增序列为这个图的度序列 (degree sequence). 例如图 1 的度序列为 $(5, 3, 3, 2, 1)$. 每个图都有一个度序列; 反之, 并非每个非递增序列都为某个图的度序列. 例如 $(5, 4, 3, 2, 1)$ 就不可能是某个图的度序列, 因为定理 1.1.3 告诉我们, 一个非递增序列要成为某个图的度序列, 必须先满足序列的元素和为偶数. 其实, 这个显然的必要条件也是充分的.

定理 1.1.5 非负整数 d_1, d_2, \dots, d_n 是某个图的所有顶点度数当且仅当 $\sum d_i$ 是偶数.

证明 \Rightarrow 由定理 1.3, 显然.

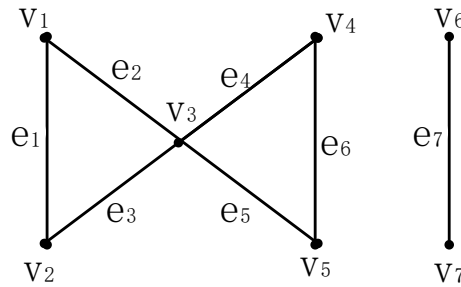
\Leftarrow 设 $V = \{v_1, v_2, \dots, v_n\}$. 显然集合 $\{v_i : d_i \text{ 是奇数}\}$ 的元素个数为偶数. 将此集合中元素两两配对, 对每个元素对, 构造一条边使其端点为此元素对. 则此时每个顶点 v_i 需要的度数是偶数 (非负), 在 v_i 处加上 $\lfloor d_i/2 \rfloor$ (表示数的下取整, 如 $\lfloor 1.5 \rfloor = 1$) 个环, 就得到以 V 为顶点集的图, 且 v_i 的度为 d_i .

定理 1.5 的证明是构造性的, 当然可以用其它方法来证明, 例如用归纳法 (对 n 或 $\sum d_i$ 进行归纳), 证明留给读者. 定理 1.5 中对度序列的刻画比较简单是因为允许使用环. 如果不允许使用环, $\sum d_i$ 是偶数的条件就不充分了. 我们在习题中给出无环图度序列的刻画. 关于简单图度序列的刻画以及更多关于度序列的内容参考[1].

定义 1.1.6 图 G 中顶点和边的交替序列 $\Gamma = v_0 e_1 v_1 e_2 \dots e_n v_n$ 称为一条 (v_0, v_n) -通道 ((v_0, v_n) -walk), 其中 v_{i-1} 和 v_i 是 e_i 的端点. v_0 和 v_n 分别称为通道 Γ 的起点 (origin) 和终点 (terminus), 其它顶点称为内点. Γ 中边的数目 n 称为通道的长度. 若起点和终点相同, 称此通道是闭的. 如果 Γ 中的边互不相同, 则称 Γ 为一条迹 (trail); 如果 Γ 中的顶点互不相同, 则称 Γ 为一条路径 (path). 起点和内点互不相同的闭通道称为圈 (cycle). 显然环是长度为 1 的圈; 一个图中有重边 \Leftrightarrow 图含有长度为 2 的圈. 若对于图中任意两个顶点 v_i 和 v_j , 都存在一

条 (v_i, v_j) -通道, 则称此图是连通的 (connected) .

在图 2 中, $v_4e_4v_3e_2v_1e_1v_2e_3v_3e_5v_5$ 是一条 (v_4, v_5) -通道, 它包含一条 (v_4, v_5) -路径 $v_4e_4v_3e_5v_5$. 这不是偶然的, 可以证明: 一条 (u, v) -通道包含一条 (u, v) -路径 (请自行验证); $v_3e_4v_4e_6v_5e_3v_2e_1v_1e_2v_3$ 是一条闭迹; $v_3e_4v_4e_6v_5e_3v_3$ 是一个圈. 注意我们常常简单的用 $e_4e_6e_5$ 或 $v_3v_4v_5v_3$ 来表示这个圈. 此图不连通, 因为不存在一条 (v_4, v_6) -通道.



定义 1.1.7 设 $G = (V, E)$, $G' = (V', E')$ 是两个图. 若 $V' \subseteq V, E' \subseteq E$, 则称 G' 是 G 的子图 (subgraph). 若 G' 是 G 的子图且 $V' = V$, 则称 G' 是 G 的生成子图 (spanning subgraph). 设 $V_1 \subseteq V$, 以 V_1 为顶点集, 以两端点全在 V_1 中的全体边为边集的 G 的子图称为 G 的导出子图 (induced subgraph), 记为 $G[V_1]$.

定义 1.1.8 图 G 的连通分支 (connected component) 是指其极大连通子图. 图 G 的割点 (cut-vertex) (割边 (cut-edge)) 是指一个顶点 (一条边) 且删除它会增加连通分支的数目.

设 $G = (V, E)$, 其中 $V = \{v_1, v_2, \dots, v_m\}, E = \{e_1, e_2, \dots, e_n\}$. 我们用 $G - v$ 和 $G - e$ 来分别表示删除点 v 和边 e 所得到的子图, 则删除顶点与边的定义为:

$$G - v_i = G[V - \{v_i\}] (i = 1, 2, \dots, m), \quad G - e_i = (V, E - \{e_i\}) (i = 1, 2, \dots, n).$$

例如图 2 有两个连通分支, 只有一个割点 v_3 , 也只有一条割边 e_7 . 图 1 没有割点, 只有一条割边 e_4 .

下面来刻画割边.

定理 1.1.9 一条边是割边当且仅当它不属于任何一个圈.

证明: 设 $e \in E(G)$, 不妨设 G 是连通的 (为什么?) .

\Rightarrow 若 e 位于某个圈中, 则不难证明 $E-e$ 连通, 这与 e 是割边矛盾, 故 e 不属于任何一个圈.

\Leftarrow 若 e 不是割边, 则 $E-e$ 连通. 设 e 的两个顶点分别为 v_1, v_2 . 由于 $E-e$ 连通, 故 $E-e$ 中存在一条 (v_1, v_2) -路径, 这条路径加上 e 就构成了一个圈.

定义 1.1.10 一个有向图 (digraph) D 是指一个二元组 (V, E) , 其中集合 V 中的元素称为顶点, 集合 E 中元素为 V 中元素组成的有序对, 称为弧或有向边 (arc or directed edge).

有向图也可以用图形表示. 例如: 图 3 中给出了 3 个有向图. 但要注意有向图中弧 (a, b) 是有方向的, 箭头从 a 指向 b . 有些概念对有向图或无向图都适用; 有些概念对有向图和无向图而言是有差异的. 本讲义中我们主要讨论无向图.

下面以一个简单的命题来结束本节. 证明命题所用的极端化方法我们将在图论中经常用到.

命题 1.1.11 如果图 G 中每一个顶点的度数至少是 2, 则 G 含有一个圈.

证明: (极端化方法¹). 设 P 是 G 中极大路径 (即不存在包含 P 的路径), u 是 P 的一个端点. 由于 P 是极大路径, 故 u 的所有相邻顶点出现在 P 中. 又由于 u 的度大于 2, 故存在 u 的相邻顶点 $v \in P$ 且 $(u, v) \notin P$. 这样边 (u, v) 以及 P 中 u 到 v 的路径就构成了一个圈.

¹ 注意本讲义讨论有限图. 极端化方法就是利用图的有限性, 当我们考虑某些极端情况时会得到一些附加的信息. 不难举例说明对于无限图, 此命题不成立.

习 题

1. 如果简单图 G 的每个顶点的度数为 2, G 一定是圈吗?

2. 给定下列各序列:

(1) $(2, 2, 2, 2, 2)$ (2) $(3, 2, 2, 1, 1)$; (3) $(2, 2, 2, 1, 1)$;

(4) $(3, 3, 3, 1, 0)$; (5) $(5, 4, 4, 3, 1)$.

以上五组数中, 那几组数可以构成简单图的度数序列?

3. 证明: 含有 n 个顶点和 k 条边的图至少有 $n-k$ 个连通分支. (提示: 添加一条边至多减少一个连通分支).

4. 证明: 如果一个图的所有顶点的度都为偶数(这样的图称为**偶图**), 则此图没有割边.

5. 对于含有 n 个顶点的简单图, 如果任意两个顶点都有边相连(即每个顶点的度为 $n-1$), 则称此图为**完全图**(complete graph), 记为 K_n . 确定 K_4 是否包含以下情况(给出例子或证明不包含).

(1) 一个不是迹的通道.

(2) 一个不是路径的闭迹.

(3). 一个不是圈的闭迹.

6. 对于图 G 和 H , 如果存在两个双射 $\theta: V(G) \rightarrow V(H)$ 和 $\lambda: E(G) \rightarrow E(H)$ 使得对于任意的 $e = (v, u) \in E(G)$, $(\theta(v), \theta(u)) = \lambda(e)$, 则称 G 和 H **同构**(isomorphism), 记为 $G \cong H$. 一个简单图 G 的**补图**(complement) G^c 也是一个简单图, 其顶点集为 $V(G)$, 且 $(u, v) \in E(G^c) \Leftrightarrow (u, v) \notin E(G)$.

(1) 证明: P_4 (n 个顶点的路径记为 P_n) 和其补图同构. 像这样和其补图同构的图称为**自补**的 (self-complementary).

(2) 证明: $G \cong H \Leftrightarrow G^c \cong H^c$.

(3) 证明简单图集合的同构关系时一种等价关系.

(4) 判断下面两个图是否同构.

7. 有向图 $D=(V, E)$ 中的 (u_1, u_n) -路径是指顶点序列 $u_1 u_2 \dots u_n$ 且 (u_i, u_{i+1}) 是有向图中的弧 ($\forall i=1, 2, \dots, n-1$). 有向图的**底图**(underlying graph)是指将边看为无向边之后得到的无向图. 有向图是**单连通**的 (unilaterally connected) 的是指 $\forall u, v \in V$, 或存在 (u, v) -路径或存在 (v, u) -路径. 有向图是**弱连通**的 (weakly connected) 是指其底图是连通的. 有向图是**强连通**的是指 $\forall u, v \in V$, 存在 (u, v) -路径和 (v, u) -路径. 考察图 3 中的有向图的连通性.
8. 证明: 如果简单图 G 的任意顶点度数至少是 k ($k \geq 2$), 则 G 包含一个长度至少为 $k+1$ 的圈 (提示: 参考命题 1.11 的证明).
9. 设图 G 有 n 个顶点且至少有 n 条边, 证明 G 含有一个圈 (提示: 用极端化方法).
10. 设 $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$ 且 $d_i \in \mathbb{N}^+$, 证明存在一个无环图 (允许有重边) 使其度序列为 d_1, d_2, \dots, d_n 当且仅当 $\sum d_i$ 为偶数且 $d_1 \leq d_2 + d_3 + \dots + d_n$ (Hakimi S.L., 1962).

1.2 树的性质

树是图论中最有用的概念之一. 作为一种特殊的图, 树在数据存储、查询, 通信, 电网分析, 化学等方面有着重要应用. 本节研究树的基本性质.

定义 1.2.1 一个森林 (forest) 是指一个无圈图. 一棵树 (tree) 是指一个连通的森林. 度为 1 的顶点称为叶子 (leaf). 若一个图的生成子图是一棵树, 则称该树是图的生成树 (spanning tree).

例 1.2.2 给西安电子科技大学的所有学生编排学号时形成一颗树. 以 01, 02, 03, ... 表示学院, 以 10, 11, 12, ... 表示入学年份为 2010, 2011, 2012, ... 以 1, 2, 3... 表示学院的班级, 以 001, 002, 003, ... 表示各班级的学生. 这一结果如下图所示, 树中的每个叶子表示一个学生. (例如: 顶点为 010 的叶子所表示的学生学号可记为 07132010, 表示该学生是 07 学院 13 级 2 班第 10 号)

下面给出树的等价刻画.

定理 1.2.3 对于 n 个顶点的图 G , 下面命题等价:

- (1) G 是连通的且无圈.
- (2) $\forall u, v \in V(G)$, G 中只有一条 (u, v) -路径且 G 无环.
- (3) G 有 $n-1$ 条边且无圈.
- (4) G 是连通的且有 $n-1$ 条边.

证明: (1) \Rightarrow (2). 由于 G 是连通的, 故 $\forall u, v \in V(G)$, G 中有一条 (u, v) -路径. 若 G 中还有一条不同的 (u, v) -路径, 则不难证明 G 中有圈 (习题 1), 与 G 中无圈矛盾, 所以 G 中只有一条 (u, v) -路径. 无圈推出无环.

(2) \Rightarrow (3). 上面证明已经指出: $\forall u, v \in V(G)$, 若 G 中只有一条 (u, v) -路径, 则 G 中无圈. 下面用归纳法证明 G 有 $n-1$ 条边. 如果 $n=1$, 结论是显然的 (由于 G 无环), 下设对于顶点数小于 n ($n \geq 2$) 的图命题成立. 对于 G 中任意一条边 (u, v) , 在图 G 中删除此边得到图 H . 由于 G 中只有一条 (u, v) -路径, 故 H 不连通. 不难证明 H 有两个连通分支且无圈, 设这两个连通分支分别为 H_1 和 H_2 . 由 (1) \Rightarrow (2) 知这两个连通分支中任意两顶点间只有一条路径, 又由于 H_1 和 H_2 的顶点数都小于 n , 由归纳假设知 $e_i = n_i - 1$ (其中 e_i, n_i 分别表示 H_i 的边和顶点数, $i=1, 2$). 又 $n = n_1 + n_2 = e_1 + e_2 + 1 + 1 = e + 1$, 证毕.

(3) \Rightarrow (4). 设 H_1, H_2, \dots, H_k 使 G 的连通分支. 由 (1) \Rightarrow (2) \Rightarrow (3) 知, 对每个连通分支 H_i 都有 $e_i = n_i - 1$ (其中 e_i, n_i 分别表示 H_i 的边和顶点数, $i = 1, 2, \dots, k$) . 故

$$\sum_{i=1}^{i=k} e_i = \sum_{i=1}^{i=k} n_i - k, \text{ 即 } e = n - k.$$

由于 $e = n - 1$, 故 $k = 1$, 即 G 连通.

(4) \Rightarrow (1). 若 G 中有圈, 则从各个圈中删除边, 直到 G 中无圈. 又因为删除圈中的边不会破坏连通性 (也即圈中的边一定不是割边, 见定理 1.1.9), 故最后得到的图 G^* 是连通的无圈图且顶点数为 n . 由于 (1) \Rightarrow (2) \Rightarrow (3), 故 G^* 有 $n - 1$ 条边. 所以 $G^* = G$ 且是无圈的.

我们在习题中给出树的其它的性质, 下面给出关于生成树的一个结果.

定理 1.2.4. G 是连通的当且仅当它有生成树.

充分性是显然的, 必要性的证明类似于定理 1.2.3 证明中的 (4) \Rightarrow (1), 故我们略去. 注意定理 1.2.4 给出了确定图 G 是否连通的方法, 即检测图是否有生成树. 我们将在 1.4 节讨论求 G 的生成树的算法.

命题 1.2.5 设 $G = (V, E)$, $A, B \subseteq E(G)$. 令 $I = \{Y \subseteq E(G) : Y \text{ 不含圈}^2\}$, 若 $A, B \in I$, $|A| < |B|$. 则存在 $e \in B - A$ 使得 $A \cup \{e\} \in I$.

证明: 考虑 $A \cup B$ 的导出子图 $G[A \cup B]$. 由定理 1.2.3, 一个图的极大无圈子图有相同的边数. 由于 $|A| < |B|$, 故 $G[A]$ 是 $G[A \cup B]$ 的无圈子图但不是极大无圈子图. 因此存在 $e \in B - A$ 使得 $A \cup \{e\}$ 不含圈.

注记 1.2.6* 我们介绍一个有趣的事实. 在线性代数中, 我们有一个类似于命题 1.2.5 的结论: 令 $X = \{a_1, a_2, \dots, a_n\}$. $I = \{Y \subseteq X : Y \text{ 中的向量线性无关}\}$, 若 $A, B \in I$, $|A| < |B|$. 则存在 $e \in B - A$ 使得 $A \cup \{e\} \in I$. 后面习题 4 给出了生成树的一个重要性质, 其实类似的结论对于向量空间中的极大线性无关组也成立, 这就启发我们这两种情况中存在某种类似的抽象结构. 这种结构我们称之为拟阵 (matroid), 是 Whitney 在 1935 年研究图

²严格的说 A 不含圈是指 A 的任意子集都不是某个圈的边集. 或者说 A 的导出子图不含圈. 注意我们前面只定义了点集的导出子图. 而 A 的导出子图 $G[A]$ 是以 A 中所有边的顶点为顶点集, A 为边集的子图

中的独立性时首先提出的（习题）。

下面来研究 n 个顶点的树的个数问题，我们将得到一个漂亮的计数公式。令 $S = \{1, 2, 3, \dots, n\}$ ，以 S 为顶点集可以构成多少棵树呢？当 $|S| = 1, 2, 3, 4, \dots$ ，我们发现树的个数分别为 $1, 1, 3, 16, \dots$ 。这样我们发现 $\tau(n) = n^{n-2}$ ($\tau(n)$ 表示 n 个顶点的树的个数)，这就是著名的 Cayley 公式。

在证明这个公式之前我们先对这里的计数问题做一说明。以 3 个顶点为例，存在 3 棵不同的树，但这三棵树是同构的，因此这里的不同并非同构意义下的不同。这三棵树的边集分别记为 $\{12, 23\}, \{13, 23\}, \{12, 13\}$ ，可以看出这三棵树的边集的标记是不同的，因此有的书上也称这里的树为标记树 (labelled tree)。最后指出以 S 为顶点的树的个数显然等于 K_n 中生成树的个数。我们用 $\tau(G)$ 表示连通图 G 的生成树的个数，因此 Cayley 公式就可表述为：

定理 1.2.7(Cayley 公式) $\tau(K_n) = n^{n-2}$

证明一 (Prüfer 编码)。令 $S = \{1, 2, 3, \dots, n\}$ 是 K_n 的顶点集，则由 S 中的元素组成的长度为 $n-2$ 的序列有 n^{n-2} 个。这样我们要证明 $\tau(K_n) = n^{n-2}$ ，只需证明 K_n 的生成树之集和这些序列之集之间存在一个双射。设 T 是 K_n 的一棵生成树，下面给 T 赋予一个长度为 $n-2$ 的序列（这个序列称为 T 的 Prüfer 编码）。

设 s_1 是 T 中顶点度为 1 的最小顶点， t_1 是 s_1 的相邻顶点。取 t_1 为 T 的 Prüfer 编码的第 1 个元素。设 s_2 是 $T - s_1$ 中顶点度为 1 的最小顶点， t_2 是 s_2 的相邻顶点。取 t_2 为 T 的 Prüfer 编码的第 2 个元素。继续这样的过程，直到我们得到 T 的 Prüfer 编码 $(t_1 t_2, \dots, t_{n-2})$ 。

我们下面举例说明这个过程，例如图 6 中的树的 Prüfer 编码为 (3344)。我们发现首先度数为 1 的顶点不出现在编码中，其次顶点在编码中出现的次数为度数减 1。

图 6

不难看出, 不同树的 Prüfer 编码不同, 证明留作习题. 最后我们说明任意一个长度为 $n-2$ 的序列 $(t_1 t_2, \dots, t_{n-2})$ 都是某棵树的 Prüfer 编码.

设 s_1 是 S 中而不在 $t_1 t_2, \dots, t_{n-2}$ 中的最小顶点, 连接 t_1 和 s_1 . 设 s_2 是 $S - \{s_1\}$ 中而不在 t_2, \dots, t_{n-2} 中的最小顶点, 连接 t_2 和 s_2 . 设 s_3 是 $S - \{s_1, s_2\}$ 中而不在 t_3, \dots, t_{n-2} 中的最小顶点, 连接 t_3 和 s_3 . 继续这样的过程得到 $n-2$ 条边 $t_1 s_1, \dots, t_{n-2} s_{n-2}$, 再连接 $S - \{s_1, s_2, \dots, s_{n-2}\}$ 中两点就得到了 K_n 的一棵生成树. 显然这棵树的 Prüfer 编码就是 $(t_1 t_2, \dots, t_{n-2})$. 这样我们就在 K_n 的生成树之集和由 S 中的元素构成的长为 $n-2$ 的序列之集之间建立了一个双射, 从而证明了 $\tau(K_n) = n^{n-2}$.

证明二(递归公式). 我们证明一个更一般的公式 $f(n, s) = sn^{n-s-1}$, 其中 $f(n, s)$ 表示 n 个顶点 s 个连通分支的森林的个数. 则 $\tau(K_n) = f(n, 1) = n^{n-2}$.

首先我们证明如下的递归公式:

$$f(n, s) = \sum_{j=0}^{n-s} \binom{n-s}{j} f(n-1, s+j-1) \quad (1)$$

令 $f(1, 1) = 1, f(n, 0) = 0 (n \geq 1)$, 设 G 是以 $\{1, 2, 3, \dots, n\}$ 为顶点集合的 s 个连通分支的森林. 假设 $1, 2, 3, \dots, s$ 是属于不同连通分支的顶点. 和 1 相邻的顶点集可以是 $\{s+1, s+2, \dots, n\}$ 的任意子集, 也即顶点 1 的度数 j 可以是 $0, 1, \dots, n-s$ 中任意一个数, 故 1 的相邻顶点集有 $\binom{n-s}{j}$ 种可能. 又因为 $G-1$ 是以 $\{2, 3, \dots, n\}$ 为顶点集合的 $s+j-1$ 个连通分支的森林, 故 $G-1$ 有 $f(n-1, s+j-1)$ 种可能. 这样就证明了(1).

现在对 n 用归纳法来证明 $f(n, s) = sn^{n-s-1}$. 当 $n=1$ 时显然成立. 假设

$$f(n-1, i) = i(n-1)^{n-i-2} (n \geq 2), \quad i = 1, 2, 3, \dots, n-1 \quad (2)$$

把(2)代入(1)得

$$\begin{aligned}
f(n, s) &= \sum_{j=0}^{n-s} \binom{n-s}{j} (s+j-1)(n-1)^{n-s-j-1} \\
&= \sum_{j=0}^{n-s} \binom{n-s}{j} (n-1)^{n-s-j} - \sum_{j=0}^{n-s-1} \binom{n-s}{j} (n-s-j)(n-1)^{n-s-j-1}
\end{aligned}$$

由二项式定理³,

$$= n^{n-s} - (n-s)n^{n-s-1} = sn^{n-s-1}.$$

注记 1.2.8 Caylay 公式给出了完全图中生成树的个数, 那么对于一般的连通图 G , 著名的矩阵树定理(matrix tree theorem)给出了 $\tau(G)$ 的计算方法, 参考 [].

注记 1.2.9 若 $f(n)$ 表示 n 个顶点的森林的个数, Takács 证明了

$$f(n) = \frac{n!}{n+1} \sum_{j=0}^{\lfloor n/2 \rfloor} (-1)^j \frac{(2j+1)(n+1)^{n-2j}}{2^j j!(n-2j)!}.$$

有趣的是, Rényi 证明了下面和常数 e 相关的渐进结果

$$\lim_{n \rightarrow \infty} \frac{f(n)}{n^{n-2}} = \sqrt{e} \approx 1.6478.$$

³ $(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}$

习 题.

1. 图 G 中有两条 (u, v) -路径, 则 G 中含圈.

2. 图 G 是树 $\Leftrightarrow G$ 连通且 G 中每一条边都是割边

\Leftrightarrow 在 G 中任意添加一条以 $V(G)$ 中顶点为端点的边都恰好构成一个圈.

3. 设 G 是一棵树. 证明: G 的顶点全部是奇数度当且仅当对所有的 $e \in E(G)$, $G - e$ 的两个连通分支都有奇数个顶点.

4. 设 G_1, G_2 是连通图 G 的两个生成树并且 $e \in E(G_1) - E(G_2)$, 则存在边 $e^* \in E(G_2) - E(G_1)$, 使得 $G_1 - e + e^*$ 是 G 的一颗生成树.

5*. 设 X 是非空有限集合, \mathbf{I} 是 X 的非空子集族且满足下面的性质:

(a) $Y \in \mathbf{I}, Z \subseteq Y \Rightarrow Z \in \mathbf{I}$.

(b) $Y, Z \in \mathbf{I}, |Y| < |Z| \Rightarrow$ 存在 $z \in Z - Y$, 使得 $Y \cup \{z\} \in \mathbf{I}$.

则称 (X, \mathbf{I}) 是一个拟阵. \mathbf{I} 中的极大元称为拟阵的基, 所有基之集记为 \mathbf{B} . 证明:

(1) 拟阵的所有基都有相同的势. 即 $\forall B_1, B_2 \in \mathbf{B} \Rightarrow |B_1| = |B_2|$.

(2) 设 $B_1, B_2 \in \mathbf{B}$, 且 $b \in B_1 - B_2 \Rightarrow$ 存在 $b^* \in B_2 - B_1 \Rightarrow \{b^*\} \cup B_1 - b \in \mathbf{B}$.

6. 设 T 是一棵偶数个顶点的树, 证明 T 恰有一个生成子图使得其中每个顶点的度数为奇数.

7. 证明: 不同树的 Prüfer 编码不同.

8. 构造以 $\{1, 2, \dots, n\}$ 为顶点集且 Prüfer 编码分别为 $(2, 3, \dots, n-2, n-1)$, $(2, 3, \dots, n-3, n-2, n-2)$, $(3, 3, 4, \dots, n-3, n-2, n-2)$ 的树.

9. 设 d_1, d_2, \dots, d_n 是正整数序列. 证明存在以 d_1, d_2, \dots, d_n 为度序列的 n 个顶点的树当且仅当 $d_1 + d_2 + \dots + d_n = 2(n-1)$. 构造一棵度序列为 $1, 1, 1, 1, 2, 3, 3$ 的树 (提示可用

Prüfer 编码).

10. 图 G 收缩(contract)边 e 是指先删除边 e , 再将边 e 的两个顶点合为一个顶点, 得到的新图记为 $G \bullet e$ (例如图 7 给出了一个收缩的例子). 证明若 e 不是环, 则

$$\tau(G) = \tau(G - e) + \tau(G \bullet e)$$

1.3 根树及其应用

我们下面讨论树在计算机学科中的一些应用, 先给出根树的定义.

每一颗树都可以按下面的方式画出来 (例如图 2.1). 每一个顶点都有层次 (level) $0, 1, 2, \dots, k$. 存在的唯一的层次为 0 的顶点称为根 (root). 所有相邻的顶点相差一个层次, 且 $i+1$ 层次上的顶点正好与一个 i 层次上的顶点相邻. 这样的树称为根树 (rooted tree). 显然指定一个顶点作为根后, 每一颗树都可以认为是根数. k 称为树高 (height). 与顶点 u 相邻且比 u 层次低的顶点称为 u 的孩子 (children), u 称为此顶点的父亲. 比 u 层次低且与 u 有路径相连的顶点称为 u 的后代 (descendant). 如果一颗树的每个顶点有不多于 m 个孩子, 则称此树为 m 元 (m -ary) 树. 如果每个顶点有 0 个或 m 个孩子, 则称此树为完全 m 元树. 2 元根树也称为二叉树 (binary tree).

下图给出了一颗高为 3 的完全二叉树.

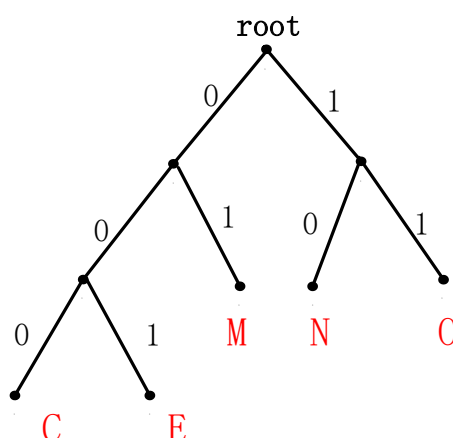


图 8

当传输数据时, 每一个字符被编码或被指定一个二进制串. 例如我们要传送 $\alpha, \beta, \gamma, \varepsilon, \sigma$ 这五个符号时就需要至少三位等长二进制串. 由于我们可能要处理

一些很大的文件且计算机的储存空间有限, 我们常常希望将字符编码成二进制串使得文件总长最小. 不妨设这五个字符在文件中出现得频率分别为 a_1, \dots, a_5 , 则如果用三位二进制串传输这一数据集合需要

$$3(\alpha a_1 + \beta a_2 + \dots + \sigma a_5) \text{ 位.}$$

如果允许使用变长度的二进制串, 则可能节省很多. 例如我们给上面 5 个字符分别编码如下:

$$\alpha: 0; \beta: 00; \gamma: 01; \varepsilon: 001; \sigma: 010$$

则传输这一数据集合需要

$$\alpha a_1 + 2(\beta a_2 + \gamma a_3) + 3(\varepsilon a_4 + \sigma a_5) \text{ 位.}$$

显然变长度的传输方式节省了传输位数, 然而却出现了问题. 如果我们收到二进制串: 001..... 则没有办法解码. 出现这样问题的原因很简单, 是因为我们在编码过程中存在某个字符的二进制串是其它字符二进制串的前面部分. 例如上例中 α 的二进制串是 β 的前面部分等等. 因此我们分不清楚 00 是表示两个 α 还是一个 β . 我们称这种情况不出现的编码为前缀编码 (prefix code). 下面我们来介绍构造最优前缀编码的算法 (Huffman 1952).

Huffman 算法: 最优前缀编码

输入: 权值 (频率或概率) a_1, a_2, \dots, a_n .

输出: 前缀码 (一颗二叉树).

思想: 权值小的字符 (或数据) 应该有较长的编码, 将权值小的数据置于二叉树的深层.

步骤: 1. 取权值最小的 a_i, a_j 为叶子, 做一个新顶点为这两个树叶的父亲且其权为 $a_i + a_j$. 从所有的权中删去 a_i, a_j 加上 $a_i + a_j$.

2. 重复 $n-1$ 次过程 1 后就得到一颗二叉树.

3. 从二叉树的树根出发, 给左边的边标记 0, 右边的边标记 1, 则树叶的编码为从树根到树叶的路径上标记的字符串.

关于这一算法最优性的证明可参考 [1] 或 [2]. 下面我们用一个例子说明求最优前缀编码的过程.

例 1.3.1 在通信中, 0, 1, 2, 3, 4, 5 出现的频率分别如下:

$$0: 35\% \quad 1: 20\% \quad 2: 15\% \quad 3: 15\% \quad 4: 5\% \quad 5: 10\%$$

求传输它们的最优前缀码.

解: 按照 Huffman 算法得到的二叉树如下图 9(a) 所示. 所以 0, 1, 2, 3, 4, 5 的最优前缀编码分别为

0: 1 1: 011 2: 010 3: 001 4: 0000 5: 0001

按比例传输上述字符 1000 个需要的二进制数字个数为

$$1000 (35\% \times 1 + 20\% \times 3 + 15\% \times 3 + 15\% \times 3 + 10\% \times 4 + 5\% \times 4) = 2450 \text{ 个.}$$

如果所有字符都用长为 3 的码字传输, 则需要二进制数字 3000 个. 所以用最优前缀编码省了 550 个二进制数字.

图 9(a) 中的树称为代码树. 注意 Huffman 算法可以产生不同的最优代码树, 如 9(b) 也是一颗最优代码树 (参考习题 6).

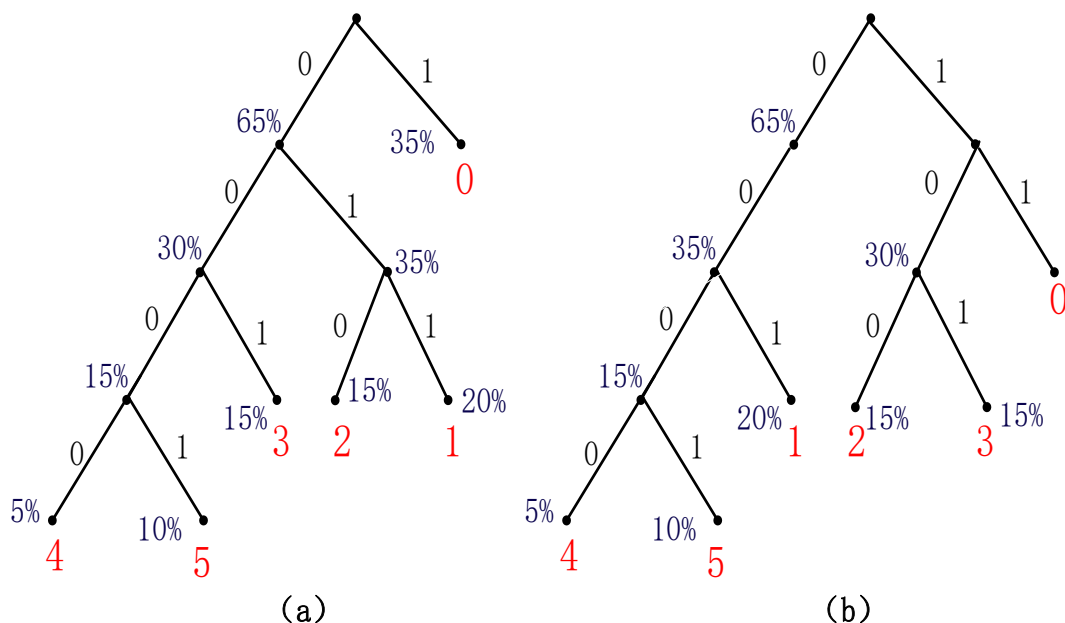


图 9

下面我们介绍根树的第二个应用. 考虑搜索 n 个数字列表中的某个数字. 最坏情况是, 该数字处于列表的最后位置, 则我们需要 n 步找到它. 下面我们用二叉搜索树储存这些数据, 并证明通过这种方法我们搜索某个数的步骤将大大降低.

一颗二叉搜索树 (binary search tree) 是一颗给每一个顶点赋值了的二叉树, 且对每一个顶点如果它有孩子, 那么它的左孩子和左孩子的后代的赋值比该顶点的赋值小; 它的右孩子和右孩子的后代比该顶点的赋值大. 图 10 中给出了一个完全二叉搜索树. 各顶点的赋值为 1, 2, ..., 7. 例如要搜索 7, 先是和 4 比较, 7 大于 4 故而和 4 的右孩子比较, 7 又大于 6, 故在和 7 比较, 这样用了 3 步就找到了. 如果按照列表 1, 2, 3, ..., 7 则需要 7 步.

假如已知一颗二叉搜索树，最坏情况下用多少步可以搜索到某个顶点取决于这颗二叉树的高度。对于 n 个顶点的二叉树，我们自然关心它的最小高度是多少。

定理 1.3.2 n 个顶点的二叉树的最小高度为 $\lceil \log_2(n+1) \rceil - 1$ 。

定理的证明我们留作习题。由此定理知对于 n 个数据，我们利用二叉搜索树搜索某个数据最多需要 $\lceil \log_2(n+1) \rceil$ 步。这比按顺序查找列表中的项要好很多，因为

$$\lim_{n \rightarrow \infty} \frac{n}{\log_2(n+1)} = \infty.$$

最后我们简单介绍根树在排序中的应用。排序问题是计算机科学中的一个基础性问题。通过比较进行排序的任意算法可以由一棵称为决策树 (decision tree) 的 (完全) 二叉树表示。我们可以通过决策树给出比较排序算法的下界。下图给出排序三个数 a, b, c 的计算机程序的决策树

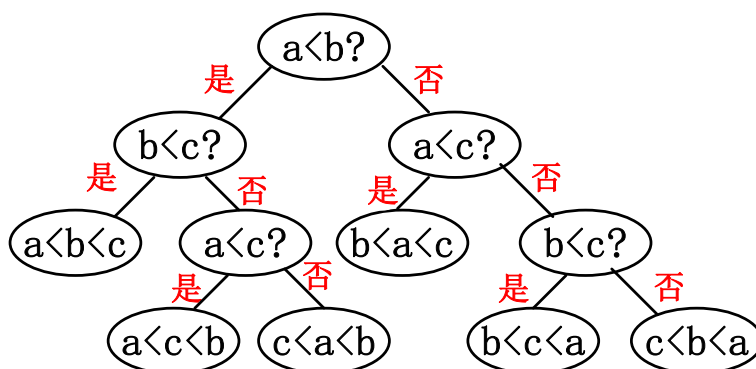


图 11

决策树中的每个叶子表示一种可能的排序结果，最坏情况下的比较次数就与决策树的高度相等。显然高度为 h 的二叉树至多有 2^h 个叶子

给 n 个数排序有 $n!$ 种可能，故要找有 $n!$ 个叶子的二叉树的最小高度是多少

$$n! \leq 2^h \Rightarrow h \geq \log_2 n! \geq \log_2 \left[n(n-1) \dots \left(\left\lceil \frac{n}{2} \right\rceil \right) \right] \geq \log_2 \left(\frac{n}{2} \right)^{\frac{n}{2}}$$

对于 $n \geq 4$ 有

$$\log_2 n! \geq \log_2 \left(\frac{n}{2} \right)^{\frac{n}{2}} = \frac{n}{2} \log_2 \frac{n}{2} \geq \frac{n}{4} \log_2 n$$

由上式我们得到了利用比较进行排序的最坏运行时间下界。确实有很多算法达到下界 $cn \log_2 n$ (c 为常数), 例如堆排序和合并排序。关于排序算法的更多内容可以参见[2].

习 题.

1. 用 10 个消息符号构成的集合, 其中各个符号的频率分别为 1, 2, 3, 4, 4, 5, 6, 7, 7, 9. 求传输这些消息符号的最优前缀编码, 并画出编码树.

2. 考虑下列数据集合: 2 个 A, 3 个 B, 5 个 C, 5 个 D.

(1) 证明 Huffman 算法可以产生不同高度的最优代码树.

(2) 如何修改 Huffman 算法使得其产生高度最小的最优代码树.

3. 证明定理 1.3.2.

(1) 证明: 树的第 l 层最多有 2^l 个顶点.

(2) 证明公式: $1 + x + x^2 + \dots + x^l = \frac{1 - x^{l+1}}{1 - x} \quad (x \neq 1).$

(3) 证明: 对 n 个顶点且高度为 l 的二叉树有 $n \leq 2^{l+1} - 1$.

(4) 证明: 存在 n 个顶点的二叉树且其高度为 $\lceil \log_2(n+1) \rceil - 1$ (考虑完全二叉树).

(5) 证明: n 个顶点的二叉树的最小高度为 $\lceil \log_2(n+1) \rceil - 1$

4. 假设在某二叉查找树中, 有 1 到 1000 之间的一些数, 现在要找出 363 这个数, 下列的节点序列中, 哪一个是不可能检查的序列([2]).

(1) 2, 252, 401, 398, 330, 344, 397, 363.

(2) 924, 220, 911, 244, 898, 258, 362, 363

(3) 925, 202, 911, 240, 912, 245, 363

(4) 2, 399, 387, 219, 266, 282, 381, 278, 363

(5) 925, 278, 347, 621, 299, 392, 358, 363.

5. 使用图 8 中的代码树, 解码下列传输数据: 00011010011110.

1.4 最小生成树和最短路径

本节我们学习图论中两个著名的优化问题: 最小生成树问题和最短路径问题.

最小生成树问题在计算机设计, 网路通信等方面有着重要的应用, 本节我们将介绍解决最小生成树问题的几个著名算法. 关于最小生成树问题的发展历史, 可参考 Graham&Hell, On the history of the minimum spanning tree problem, Annals of the History of Computing, 7(1985) 43-57.

加权图是各边都标有数值 (称为边的权值, 我们只考虑非负实数情形) 的图. 一个图的权是图中各边的权之和. 最小生成树问题就是给定一个加权连通图, 寻找一个权值最小的生成树. 下面我们给出解决这一问题的算法.

Kruskal 算法: 求加权连通图的最小生成树

输入: 一个加权连通图 G (n 个顶点).

输出: 图的权值最小的生成树.

思想: 从权值最小的边开始, 通过不断地增加边且不形成圈来扩张, 最后形成树. 先以权值递增的顺序排列各边, 权值相同的边可以任意排序.

步骤: 1. 将各边按权值递增的顺序排序, 设 T 为空集.

2. 检查排序列表中未被检查的第一条边, 并将其加入 T 中当且仅当它不与 T 中的其它边形成圈. 如果这条边加入到 T 中则进入步骤 3, 否则重复 2.

3. 如果 T 有 $n-1$ 条边则停止, 输出 T , 否则进入 2.

下面先举例说明 Kruskal 算法怎么实现. 对于图 12 中的图 G , 我们来按照 Kruskal 算法找出 G 的一棵生成树.

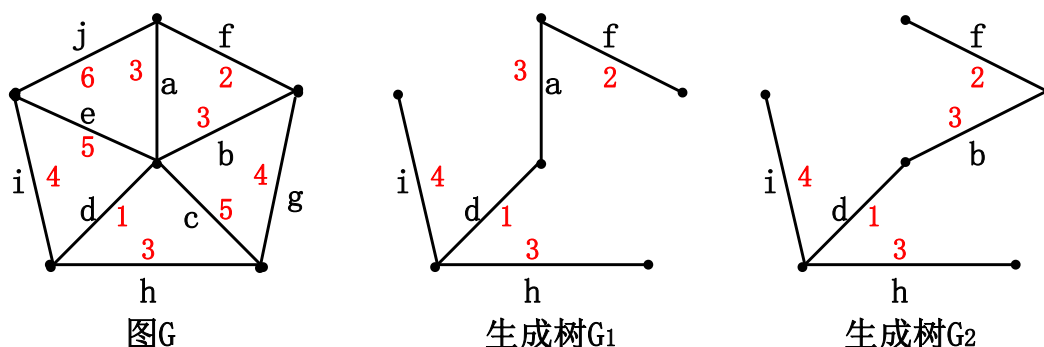


图 12

1. 给各边排序: $d, f, a, b, h, i, g, c, e, j$

2. 先将 d 加入 T , 再加入 f (因为 f 和 d 不构成圈). 这时注意 a, b, h 的权是相同的, 可以按任意顺序考虑这三条边. 不妨先考虑 a , 它和 T (这时

$T = \{d, f\}$) 不构成圈. 故将 a 加入 T . 同样接下来可以先考虑 b , 也可以先考虑 h . 如果先考虑 b , 注意 b 加入到 T 中就形成圈了 (此时 $T = \{d, a, f\}$), 故 b 不能加入到 T 中. 接着考虑 h 像这样一直下去就得到图 G 的一棵生成树 G_1 . 注意 G_2 是图 G 的另一棵最小生成树, 换句话说, 一个加权连通图的最小生成树不是唯一的. 下面我们来证明 Kruskal 算法的正确性.

定理 1.4.1 (Kruskal 1956) 在加权连通图中, Kruskal 算法构造出一棵权值最小的生成树.

证明: 考虑任意一条边时 T 是无圈图, 下面说明算法一定可以停止. 如果在考虑最后一条边后得到 T . 首先 T 一定是连通的. 因为若 T 不连通, 则一定有一条连接 T 的连通分支的边未加入到 T 中, 这与考虑了所有边矛盾. T 连通且无圈, 故 T 是树.

令 T 是由算法得到的树, 而 T^* 是权值最小的一棵生成树. 下设 $T \neq T^*$. 令 e 是选择 T 的过程中第一条位于 T 中而不在 T^* 中的边. 将 e 添加到 T^* 中就构成一个圈. 由于 T 中无圈, 故此圈中有一边 $e^* \notin T$, 考虑生成树 $T^* + e - e^*$ (为什么是生成树?).

由于在选择 e 时, e 和 e^* 均是可选的 (为什么?), 故 e 得权值不大于 e^* 的权值. 因此生成树 $T^* + e - e^*$ 的权值不超过 T^* 的权值. 和 T^* 相比, $T^* + e - e^*$ 有更多的边与选择 T 的过程开始时选用的边保持一致. 重复这样的过程, 就得到与 T 相同的最小生成树, 证毕.

Kruskal 算法是一种贪心算法 (greedy algorithm). 所谓贪心算法是指每一步的选择都是局部最优的, 期望通过所做的局部最优选择产生一个全局最优选择. 一般来说, 贪心算法得到的解不一定是最优的, 但对最小生成树问题来说, Kruskal 算法得到的生成树却是最优的. 求加权连通图的最小生成树问题有许多算法, 下面给出另外一个著名的算法: Prim 算法, 这也是一种贪心算法, 其最优性的证明留作习题.

Prim 算法: 求加权连通图的最小生成树

输入: 一个加权连通图 G (n 个顶点).

输出: 图的权值最小的生成树.

思想: 从权值最小的边开始, 通过不断地增加边且不形成圈来扩张, 最后形成树. 先以权值递增的顺序排列各边, 权值相同的边可以任意排序.

步骤: 1. 设置集合 T 为空集, 选出图的任意一个顶点放入 T 中, .

2. 把一个顶点在 T 中, 一个顶点不在 T 中的所有边中权值最小的边加入 T 中. 如果权值最小的边有多条则任选其一.

3. 如果 T 有 $n-1$ 条边则停止，输出 T ，否则进入 2.

设 $G = (V, E)$ 为简单连通图，且给各边赋予非负权值，求 G 中顶点 u 到 v 的最短路径的长度

下面介绍解决此问题的 Dijkstra(1959) 算法

先介绍此算法的要求

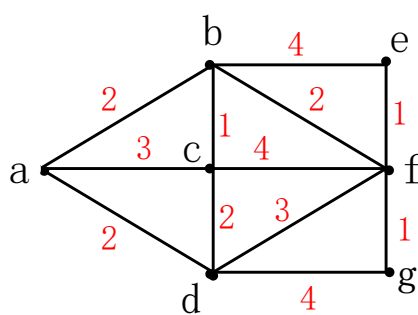
(1) 把 V 分成两个子集 S 和 T . 开始时 $S = \{u\}$, $T = V - S$, 然后不断扩充 S , 直到 $v \in S$.

(2) 对于 T 中每一元素 x 计算 $t(x)$. 根据 $t(x)$ 的值找出 T 中距 u 最短的顶点 z .

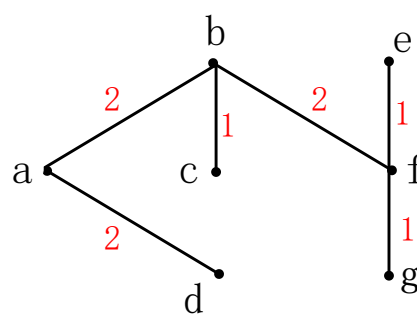
(3) 把 z 加入到 S 中. 重复 (2), 直到 $v \in S$.

在每次迭代中, 若 $z \in \{x \in T \mid t(x) = \min_{x \in T} \{t(x)\}\}$, 则 u 到 t 的最短路径长度恰为 $t(z)$, 在初始阶段 $S = \{u\}$, 则 $\forall x \in T$, $t(x) = w(ux)$. $w(ux)$ 表示边 ux 的权值; 若 u 和 x 间不存在边, 则令 $w(ux) = \infty$, 在之后的每次迭代中用来更新 $\min\{t(x), t(u) + w(ux)\}$ 来更新 $t(x)(u \in S)$, 直到 $v \in S$.

例1. 加权连通图如图所示, 求 a 到各个顶点的最短路径长度



图G



图G₁

重复次数	S	$t_i(b)$	$t_i(c)$	$t_i(d)$	$t_i(e)$	$t_i(f)$	$t_i(g)$
开始	{a}	2	3	2	∞	∞	∞
1	{a, b}		3(①)	2(②)	6	4	∞

2	{a, b, d}		3(③)		6	4	6
3	{a, b, d, c}				6	4	6
4	{a, b, d, c, f}				5		5
5	{a, b, d, c, f, e}						5

其中③代表的是 $\min\{t_1(c), t_1(b) + w(bc)\} = \min\{3, 3\}$

②代表的是 $\min\{t_1(d), t_1(b) + w(bd)\} = \min\{2, 2 + \infty\}$

③代表的是 $\min\{t_2(c), t_2(d) + w(cd)\} = \min\{3, 4\}$

习题.

1. 修改 Kruskal 算法使得可以寻找最大生成树.
2. 如果加权连通图中各边的权值都不相同, 是否可能有多棵最小生成树?
3. 证明 Prim 算法确实可以找到一棵最小生成树.
4. 证明对于一个加权连通图来说, Kruskal 算法在选择下一条边时, 如果它面临多条权值相同的边, 则无论怎么选择, 最小生成树中边的权值构成的序列 (按递增次序) 是唯一的.
5. 设 T 是加权连通图 G 的一棵最小生成树, 证明 G 中任意圈均有权值最大的一条边不在 T 中.
6. 对图3.1中的图 G , 以中间的顶点开始利用 Prim 算法寻找一颗最小生成树.

第四节 加权连通图上网最短路径问题

设 $G = (V, E)$ 为简单连通图, 且给各边赋予非负权值, 求 G 中顶点 u 到 v 的最短路径的长度

下面介绍解决此问题的 Dijkstra(1959) 算法

先介绍此算法的要求

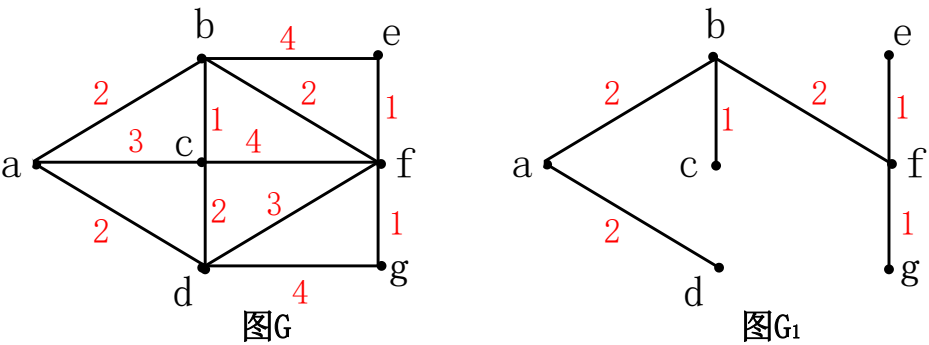
(1)把 V 分成两个子集 S 和 T . 开始时 $S=\{u\}$, $T=V-S$, 然后不断扩充 S , 直到 $v\in S$.

(3)对于 T 中每一元素 x 计算 $t(x)$. 根据 $t(x)$ 的值找出 T 中距 u 最短的顶点 z .

(3)把 z 加入到 S 中. 重复 (2), 直到 $v\in S$.

在每次迭代中, 若 $z\in\{x\in T|t(x)=\min_{x\in T}\{t(x)\}\}$, 则 u 到 t 的最短路径长度恰为 $t(z)$, 在初始阶段 $S=\{u\}$, 则 $\forall x\in T$, $t(x)=w(ux)$. $w(ux)$ 表示边 ux 的权值; 若 u 和 x 间不存在边, 则令 $w(ux)=\infty$, 在之后的每次迭代中用来更新 $\min\{t(x), t(u)+w(ux)\}$ 来更新 $t(x)(u\in S)$, 直到 $v\in S$.

例1. 加权连通图如图所示, 求 a 到各个顶点的最短路径长度



重复次数	S	$t_i(b)$	$t_i(c)$	$t_i(d)$	$t_i(e)$	$t_i(f)$	$t_i(g)$
开始	{a}	2	3	2	∞	∞	∞
1	{a, b}		3(①)	2(②)	6	4	∞
2	{a, b, d}		3(③)		6	4	6
3	{a, b, d, c}				6	4	6
4	{a, b, d, c, f}				5		5
5	{a, b, d, c, f, e}						5

其中①代表的是 $\min\{t_1(c), t_1(b)+w(bc)\}=\min\{3,3\}$

②代表的是 $\min\{t_1(d), t_1(b) + w(bd)\} = \min\{2, 2 + \infty\}$

③代表的是 $\min\{t_2(c), t_2(d) + w(cd)\} = \min\{3, 4\}$

二部图

定义：图 $G = (V, E)$ 称为二部图(bipartite graph)，如果 V 是两个互不相交的集合 V_1, V_2 的开集，且 V_1 和 V_2 中的顶点互不相邻。这样的二部图也常称为 (V_1, V_2) -二部图。

定义：图 G 的匹配是由 G 中没有公共顶点构成的集合，与匹配 M 中的边关联的顶点称为是被 M -浸润的(saturated by M)，其余的顶点称为未被 M -浸润的(M -unsaturated)。图 G 的一个完美匹配(perfect matching)是浸润的所有顶点的匹配。图 G 的边数最多的匹配称为一个最大匹配(maximum matching)。

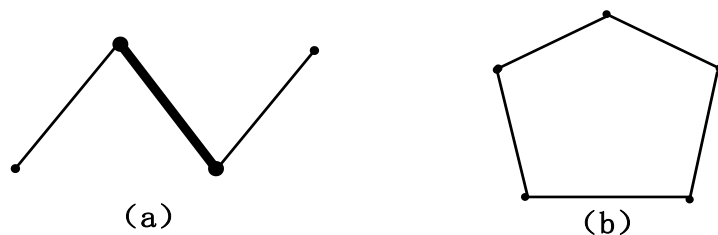


图1

例如在上图中，粗边给出了一个匹配 M_1 ，显然两条细边给出了一个最大匹配 M_2 。

定义：设 M 是图 G 的一个匹配。如果路径 P 的边交替出现在 M 和不出现在 M 中，则称 P 是一条 M -交错路径 (M -alternating path)。两个顶点都未被 M -浸润的交错路径称为 M -增广路径 (M -augmenting path)。

在上例中存在 M_1 -增广路径， M_2 是最大匹配，而不存在 M_2 -增广路径，这不是偶然的。因为可以让（留作习题）：图 G 的一个匹配 M 是最大匹配 $\Leftrightarrow G$ 中无 M -增广路径。

定义：图 G 的一个顶点覆盖 (covering) 是一些顶点构成的集合 $\kappa \subseteq V(G)$ ，使得 G 的任何一边都有一个顶点含于 κ 。一个顶点覆盖 κ 称为最小顶点覆盖，是指不存在覆盖 κ' ，使得 $|\kappa'| < |\kappa|$ 。

设 κ 是 G 的一个顶点覆盖， M 是 G 的一个匹配，显然 $|\kappa| \geq |M|$ 。我们关心对于最大匹配的最小顶点覆盖来说，等式是否成立。在图 1(a) 中，等式成立，而图 1(b) 中最小顶点覆盖大小为 3，而最大匹配大小为 2。注意图 1(a) 为二部图，图 1(b) 为有 5 条边的圈，从而不是二部图（可以一个图 G 是二部图 $\Leftrightarrow G$ 中不含奇数边的图，证明留作习题）。对于二部图，我们有下面一般的结论：

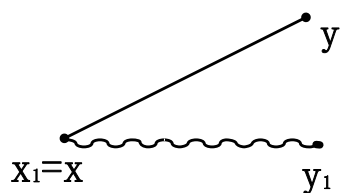
定理：设 G 是 (X, Y) -二部图，则 G 的最大匹配的大小等于 G 的最小顶点覆盖的大小 (könig 1931)。

证明：设 M 是 G 的最大匹配，而 Q 是 M 的最小顶点覆盖，要证 $|M| = |Q|$ 。

显然 $|Q| \geq |M|$ ，故只需证明存在 G 的 $|M|$ 个顶点的覆盖（则 $|M| \geq |Q|$ ），对于 M 中每一条边，如果存在未被 M -浸润的 X 中顶点出发的交错路径可达这条边，则选择此边在 Y 中的顶点；否则选择此边在 X 中的顶点，这样就选了 $|M|$ 个顶点，记为 U 。

设 $xy \in E$ ， $x \in X, y \in Y$ ，只需证明 x 或 $y \in U$ ，或 $xy \in M$ ，则由 U 的定义得

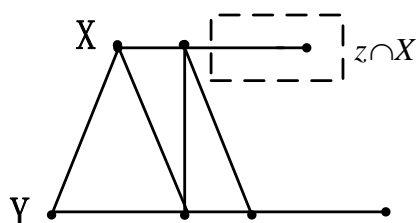
证. 下证之: 设 $xy \notin M$. 又由 M 是最大匹配, 故 $\exists x_1 y_1 \in M$ (其中 $x_1 \in X, y_1 \in Y$) 且 $x = x_1$ 或 $y = y_1$. 若 $y = y_1$ (此时 $x \notin M$), 由于 xy 是 M -交错路径, 故 $y \in U$. 下设 $x = x_1$, 如果 $x \notin U$, 则 $y_1 \in U$, 由 U 的定义: 某条交错路径可达 y_1 . 则存在交错路径 P' 可达 y ; 或 Py (若 $x_1 \in P$); 或 $Py_1 x_1 y$. 这样就出现了 M -增广路径, 与 M 是最大匹配矛盾, 故 $x \in U$.



对于 (X, Y) -二部图, 若存在一个浸润 X 的匹配, 则显然 $\forall \kappa \subseteq X$, 至少在 Y 中存在 $|\kappa|$ 个顶点与 κ 中的顶点相邻. 我们用 $N(\kappa)$ 表示与 κ 中顶点相邻的顶点构成的集合, 下面的定理说明 “ $\forall \kappa \subseteq X, N(\kappa) \geq |\kappa|$ ” 这个显然的必要条件也是充分的

定理(1935): (X, Y) -二部图中存在浸润 X 的匹配 $\Leftrightarrow \forall \kappa \subseteq X, N(\kappa) \geq |\kappa|$.

证明: “ \Leftarrow ” 由 König 定理, 只需证明对每个顶点覆盖 z , 有 $|z| \geq |X|$. 令 $s = X - z \cap X$, 则 s 的点都不在 X 中, 因此 $N(s)$ 中的点都在 z 中 (由顶点覆盖定义), 故 $|z| \geq |z \cap X| + |N(s)| \geq |z \cap X| + |s| = |X|$, 证毕.



图的连通性

因为连通与否与图是否含环无关, 故本小节假定所有图都不含环, 且 $n(G) > 1$.

定义 1: 图 G 的一个点割(vertex cut)是一个集合 $S \subseteq V(G)$, 使得 $G - S$

的连通分量多于一个 G 的连通度(connectivity), $\kappa(G)$ 是使得 $G-S$ 不连通或只有一个顶点的顶点集合 S 大小的最小值. 如果 G 的连通度最少是 κ , 则称 G 是 κ -连通的 (κ -connected).

由定义, 显然可知:

① 连通图都是 1-连通的;

② G 是不连通的 $\Leftrightarrow G$ 的连通度为 0;

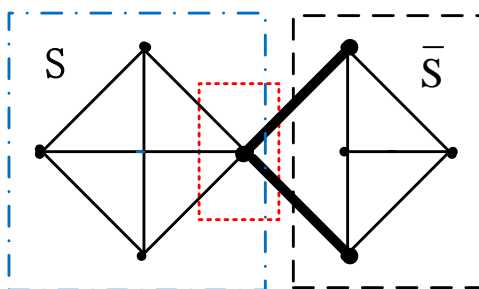
③ 顶点数大于 2 的图的连通度为 1 \Leftrightarrow 它是连通的且有一个割点.

若图 G 的连通度为 κ , 则 $\delta(G) \geq \kappa$, 故 G 中至少有 $\left\lceil \frac{\kappa n}{2} \right\rceil$ 条边 (见习题 1).

我们关心是否可以给出 n 个顶点的 κ -连通图且有 $\left\lceil \frac{\kappa n}{2} \right\rceil$ 条边 (即下界是否可以取到). 习题 1 给出了肯定的回答.

定义 2: 图 G 中的边割(edge cut)是一顶点在 S 中, 一顶点在 $V(G)-S$ 中的 G 中所有边构成的集合, 记为 $[S, \bar{S}]$ ($S \subseteq V(G)$). 若使得 $G-[S, \bar{S}]$ 不连通的 $[S, \bar{S}]$ 边数最小值为 κ , 则称 G 是 κ -边连通的, κ 称为 G 的边连通度, 记为 $\kappa'(G)$.

在下图 G 中粗线标出的边割是 G 的最小边割, 因此 $\kappa'(G) = 2$, G 是 2-边连通的. 图 G 中还标出了一个只含一个顶点的点割, 故 G 是 1-连通的.



定理 3 (Whitney 1932): 设 G 是简单图, 则 $\kappa(G) \leq \kappa'(G) \leq \delta(G)$.

证明: 设 $d(v) = \min\{d(x) : x \in V(G)\}$, 即 $d(v) = \delta(G)$, 则与 v 关联的所有边构成一个边割, 故 $\kappa'(G) \leq \delta(G)$, 下证 $\kappa(G) \leq \kappa'(G)$.

显然 $\kappa(G) \leq n(G) - 1$, 设 $[S, \bar{S}]$ 为 G 的最小边割, 若 S 中的顶点与 \bar{S} 中的

顶点都邻接, 则 $|[S, \bar{S}]| = |S||\bar{S}| \geq n(G) - 1 \geq \kappa(G)$, 命题得证.

下设存在 $x \in S, y \in \bar{S}$. 则 x, y 不相邻, 构造集合 T : T 包含 \bar{S} 中 x 的相邻顶点;
 T 包含 $S - \{x\}$ 中的所有与 \bar{S} 中顶点有相邻顶点的顶点 (或
 $T = \{v \in \bar{S} : xv \in E\} \cup \{v \in S - \{x\} : \text{存在 } u \in \bar{S} \text{ 使得 } vu \in E\}$). 因为每条 x, y
 路径都通过 T , 因此 T 是一个点割, 故 $|T| \geq \kappa(G)$. 在 $[S, \bar{S}]$ 中选 $|T|$ 条边: $\forall v \in T$,
 若 $v \in \bar{S}$, 则选边 xv ; 若 $v \in S - \{x\}$, 则任意选取一条边 $vu \in [S, \bar{S}]$, 这样选取
 的 $|T|$ 条边都是不同的, 因此

$$\kappa'(G) = |[S, \bar{S}]| \geq |T| \geq \kappa(G)$$

下面给出 2-连通图的特征.

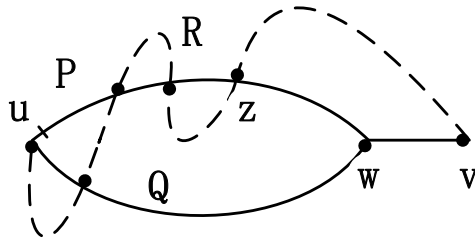
定理 4(Whitney 1932): 图 $G(n(G) \geq 3)$ 是 2-连通的 $\Leftrightarrow \forall u, v \in V(G)$, 在
 G 中存在内部不相交的 (internally-disjoint) u, v -路径 (即两条路径没有公
 共的内顶点).

证明: “ \Leftarrow ” 删除一个顶点不能使一对任意顶点不可达, 故 G 是 2-连通的.

“ \Rightarrow ” 对 $d(u, v)$ 用数学归纳法证明.

$d(u, v) = 1$, $G - uv$ 是连通的 (因为 $\kappa'(G) \geq \kappa(G) = 2$). $G - uv$ 中的 u, v -
 路径与边 uv 构成了内部不相交的两条 u, v -路径.

假设 $d(u, v) \leq \kappa - 1$ 时命题成立, 下证 $d(u, v) = \kappa$ 时命题也成立.



令 w 是某条最短 u, v -路径上 v 的前一顶点, 则 $d(u, v) = \kappa - 1$. 由归纳假
 设, G 有内部不相交 u, w 路径 P, Q . 若 $v \in V(P) \cup V(Q)$, 则在圈 $P \cup Q$ 上可

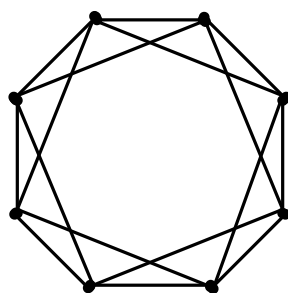
以找两条内部不相交路径. 若 $v \notin V(P) \cup V(Q)$, 由于 G 是 2-连通的, 故 $G-w$ 连通, 所以 $G-w$ 中含有一条 u, v -路径 R . 若 R 不含 P 或 Q 的内部顶点, 则完成了证明. 如若不然, 不妨设 R 与 P 的内部顶点相交, 设 z 是这些交点中在 P 上与 v 最近的一个顶点, 则 P 上的 u, z -路径合并 R 上的 z, v -路径就得到一条与 $Q \cup wv$ 内部不相交路径

练习中给出 2-连通图的其它特征. 定理 4 可以推广到一般的 κ -连通图. 证明较繁, 我们这里略去, 有兴趣的读者可参见 D. B. West, Introduction to Graph Theory, 2nd 2001. 或 J. A. Bondy, U. S. R. Murty, Graph Theory with Applications, 1976.

习题.

1. 图 G 的连通度为 κ 且 $|G| = n$, 则 G 至少有 $\left\lceil \frac{\kappa n}{2} \right\rceil$ 条边.

2. 证明下图中 $\kappa(G) = 4$, 从而满足 $|E(G)| = \left\lceil \frac{\kappa n}{2} \right\rceil$



3. 设 $|V(G)| \geq 3$, 则 G 是 2-连通的 $\Leftrightarrow G$ 是连通的且 G 无割点

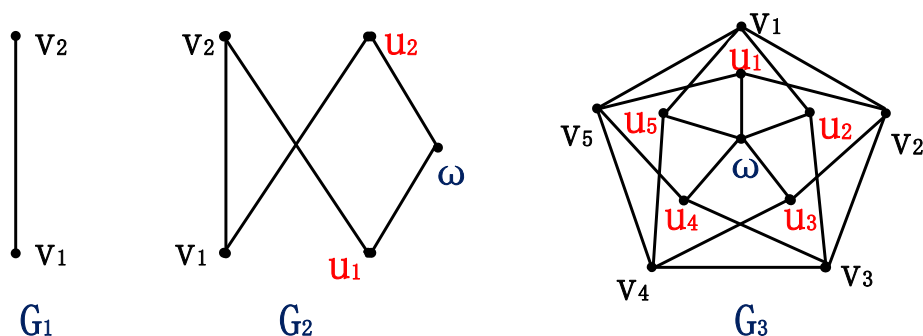
$\Leftrightarrow \forall x, y \in V(G)$, 存在经过 x, y 的环

$\Leftrightarrow \delta(G) > 1$ 且 G 的每一对边均位于一个

第五节 着色问题

定义 1. 图 G 的顶点着色 (Vertex Colouring) 是给每个顶点赋予一种颜色, 使得相邻顶点的颜色不同, 给图 G 进行顶点着色需要的颜色的最小值称为 G 的色数 (Chromatic Number), 记为 $\chi(G)$. 若 $\chi(G) = \kappa$, 则称 G 是 κ -色的 (κ -chromatic); 若 $\chi(G) \leq \kappa$, 则称 G 是可 κ -着色的 (κ -colourable).

例 1. 在下图中 $\chi(G_1) = 2$, $\chi(G_2) = 3$, $\chi(G_3) = 4$. 注意 G_1 即为 K_2 , G_2 即为 K_5



下面讨论 $\chi(G)$ 的上界和下界, 先给出下面的定义.

定义 2. 图 G 中两两不相邻的顶点组成的集合称为独立集 (Independent Set), 用 $\alpha(G)$ 来表示 G 中最大独立集的元素个数. 图 G 中两两相邻的顶点构成的集合称为团 (Clique), 用 $\omega(G)$ 表示 G 中最大团的元素个数.

注记:

(1) 显然, G 是可 κ -着色的当且仅当 $V(G)$ 是 κ 个独立集的并. 特别的, G 是可 2-着色的当且仅当 G 是二部图.

(2) $\chi(G) \geq \omega(G)$ 且 $\chi(G) \geq \frac{n(G)}{\alpha(G)}$ (习题 1).

(3) 对于完全图 K_n 有: $\chi(K_n) = \omega(K_n) = n$. 但一般来说 $\chi(G)$ 可以远大于

$\omega(G)$. 下面介绍一种构造任意色数的三角形无关图 (即不含 K_3 为子图) 的方法, 这种方法归功于 Mycielski, 1955.

定义 3 (Mycielski 构造). 由简单图 G 产生一个以 G 为子图的简单图 G' . 从顶点集为 $\{v_1, v_2, \dots, v_n\}$ 的图 G 开始添加顶点 $\{u_1, u_2, \dots, u_n\}$ 和 ω ; 添加边, 使得 u_i 与 $N_G(v_i)$ 中的顶点相邻, ω 与 $\{u_1, u_2, \dots, u_n\}$ 中的所有顶点相邻.

例 1 中, 以 2-色图 G_1 开始, 进行二次 Mycielski 构造, 分别得到 3-色图 G_2 和 4-色图 G_3 . 下面的定理告诉我们可以构造一个色数任意大的图 G 且 $\omega(G) = 2$.

定理 1 (Mycielski 1955). 由一个 κ -色三角形无关图 G , Mycielski 构造可得到一个 $\kappa+1$ -色三角形无关图 G' .

定义 4. 图 G 称为 κ -临界的 (κ -critical), 如果 $\chi(G) = \kappa$ 且 $\chi(G-v) = \kappa-1$

($\forall v \in V(G)$). 也即去掉 G 的任何一个顶点会使 G 的色数减少.

下面介绍 κ -临界图的一个重要性质.

定理 2. 如果 G 是 κ -临界的, 则 $\delta(G) \geq \kappa-1$.

证明: 令 G 是 κ -临界的且 $\delta(G) \leq \kappa-2$. 设 $v \in V(G)$ 且 $d(v) = \delta(G)$. 由于 G 是 κ -临界的, 故 $\chi(G-v) = \kappa-1$.

由于 $d(v) \leq \kappa-2$, 故在对 $G-v$ 着色的 $\kappa-1$ 种颜色中, 存在一种颜色没有被 v 的至多 $\kappa-2$ 个邻接点使用, 将这种颜色对 v 着色就得到 G 是 $\kappa-1$ -着色的. 与 $\chi(G) = \kappa$ 矛盾.

推论①: 如果 $\delta(G) = \kappa$, 则 G 至少有 κ 个顶点的度数不小于 $\kappa-1$.

证明: 如果 $\delta(G) = \kappa$, 则 G 就有一个 κ -临界子图 H (删除 G 中不影响 $\chi(G)$

的顶点,直到不能继续为止). 由于 $\chi(H) = \kappa$, 故 H 至少有 κ 个顶点. 由定理 2, $\delta(H) \geq \kappa - 1$, 所有 H 至少有 κ 个顶点的度数不小于 $\kappa - 1$, 即这 κ 个顶点在 G 中的度数不小于 $\kappa - 1$.

推论②: $\chi(G) \leq \Delta(G) + 1$.

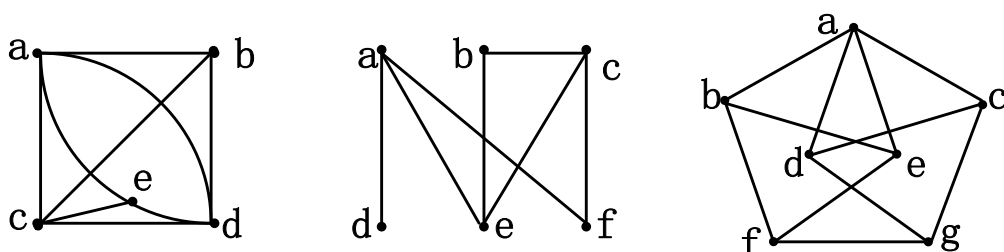
证明: 假设 $\chi(G) > \Delta(G) + 1$, 即 $\chi(G) - 1 > \Delta(G)$, 与定理 2 矛盾.

上述推论中的等号在 G 是完全图或奇环时成立, 除这两类图之外, 我们有下面的结论.

定理 3 (Brooks, 1941). 如果 G 是连通图, 并且 G 既不是完全图也不是奇环, 则 $\chi(G) \leq \Delta(G)$.

习题:

1. 证明: $\chi(G) \geq \omega(G)$ 且 $\chi(G) \geq \frac{n(G)}{\alpha(G)}$
2. 下列各图是否可以 3-着色? 确定它们的色数.



3. 新学期安排补考, 下表是上学期考试不及格的情况. “ \times ”表示某门课不及格.

学生	数分	高代	解析几何	英语
张三	\times	\times		
李四		\times		\times
王五		\times	\times	
赵六	\times	\times	\times	
陈七			\times	\times

问至少需要安排几场考试, 使得这五个同学参加完所有的考试 (注: 每场考试一个学生只能考一门, 但考场中的学生可以考不同的科目)

4. 如果 G 的度序列为 $d_1 \geq d_2 \geq \dots \geq d_n$, 则

$$\chi(G) \leq 1 + \max_{i \in \{1, \dots, n\}} \min \{d_i, i-1\}.$$

5. 图 G 的边着色是指将颜色赋予 G 的边, 如果相邻边的颜色不同, 则称这种边着色为正常边着色 (proper edge colouring). 边色数记为 $\chi'(G)$, 是对 G 进行正常边着色所需的最小颜色数量.

①证明: $\chi'(G) \geq \Delta(G)$. 举例说明某些图可以取到下界.

②证明: 对于完全图 K_{2n-1} , $\chi'(K_{2n-1}) = 2n-1$; 对完全图 K_{2n} , 有 $\chi'(K_{2n}) = 2n-1$ (因此对于任意完全图 K_n , 有 $\Delta(K_n) = n-1 \leq \chi'(K_n) \leq \Delta(K_n) + 1 = n$. 这个结果不是偶然的, 因为更强的一个结果 (Vizing 定理) 是: 对任意图 G , $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$)

③证明: 如果 G 是二部图, 则 $\chi'(G) = \Delta(G)$ (Konig, 1916).

第二章

第一节 可平面图

定义 1. 一个图 G 可以图示在平面上, 使得任何两边除了端点外无公共顶点 (即不叉), 则称图 G 是可平面图 (planar graph). G 的这种图示称为 G 的一个平面嵌入 (planar embedding). 一个平面图是指一个可平面图的一个特定的平面嵌入. 平面图将平面分成许多区域, 这些区域称为面.

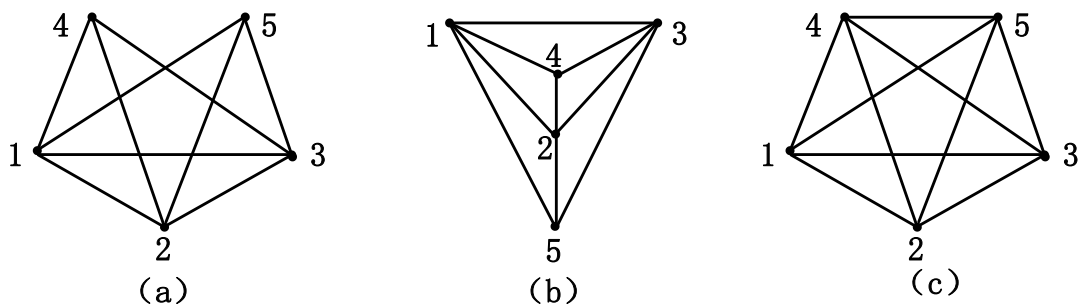
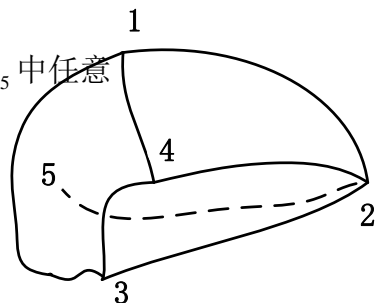


图1

例 2. 图 1(a) 是可平面图, 因为图 1(b) 给出了它的一个平面嵌入. 图 1(c) (即 K_5) 比图 1(a) 多一条边, 但它不是可平面图. 下面我们来证明这一点.

命题 3. K_5 不是可平面图.

证明: 反证法. 假设 K_5 有一个平面嵌入 K'_5 . 由于 K'_5 中任意



两个顶点相邻，故存在圈 123（不妨设 κ'_5 的顶点为 $\{1, 2, 3, 4, 5\}$ ）。

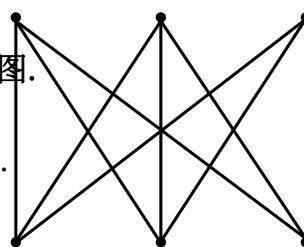
圈将平面分成了两个部分。不妨设 4 位于此圈的内部，则 124, 234, 134 将圈的内部部分成了三个区域。5 位于这四个区域中的其中一个区域内部，不妨设 5 在 123 的内部，则边 52 与圈 134 相交，这与 κ'_5 是 κ_5 的平面嵌入矛盾，故 κ_5 不是可平面图。

注：1. 上面的证明用到了平面上简单闭合曲线将平面分成两个区域（内部和外部）这个看似显然的结论。然而证明这个结论却是非常困难的，要用到代数拓扑的知识。

2. 用类似的方法可以证明 $\kappa_{3,3}$ 不是可平面图。

下面将看到由 κ_5 和 $\kappa_{3,3}$ 可以给出平面图的特征刻画。

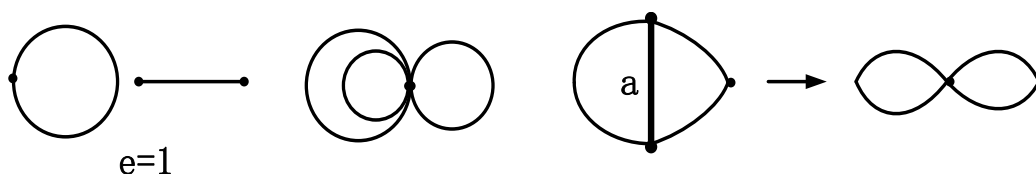
其中 $\kappa_{3,3}$ 如右图所示。



$\kappa_{3,3}$ 图

定理 4 (Euler 1758). 如果连通的平面图 G 恰好有 n 个顶点， e 条边和 f 个面，则 $n-e+f=2$ 。

证明 1：对边数 e 进行归纳。当 $e=1$ 时，因 G 是连通的，故有两种可能：即 G 是环或一条边，故 $n-e+f=2$ 成立。



假设 $e=\kappa$ 时， $n-e+f=2$ 成立。当 $e=\kappa+1(\geq 2)$ 时，找一条两个顶点的边，有两种情况：①. 不存在这样的边，即每条边都是环。由于 G 是连通的，故 G 为只有一个顶点、多个环的图，显然公式成立。②. 存在有两个顶点的边，设为 a ，我们把 a 删除，将 a 的两顶点用一新顶点代替，使得与 a 的两端点关联的边也与新顶点关联（这样的操作称为边 a 的收缩 (contracting)，记为 $G \bullet a$ ），注意收缩不会改变面的个数，但顶点数和边数各减少 1，对于 $G \bullet a$ ，公式成立。由于 G 的边数、顶点数各比 $G \bullet a$ 增加 1，而面数不变，故当 $e=\kappa+1$ 时，公式也成立。

证明 2. 设 T 为 G 的生成树，则 $e=n-1$ 且 $f=1$ ，故对 T 公式成立。 G 可以通过对 T 不断加边获得。每增加一条边将产生一个新面，即 f 和 e 值都增加 1。这使得 $n-e+f=2$ 保持不变。有限步后可从 T 得到 G ，即对于 G ，公式成立。由欧拉公式，可以得到下面的命题：

命题 5. 设 G 是至少有 3 个顶点的简单平面图，则 $e(G) \leq 3n(G) - 6$ ，若 G 是

三角形无关的（即 κ_3 不是 G 的子图），则 $e(G) \leq 2n(G) - 4$.

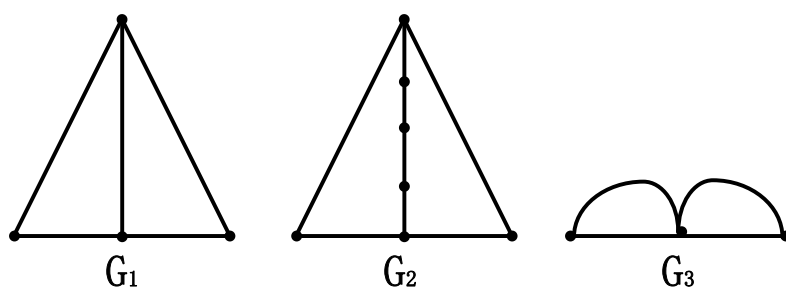
证明：每个面至少需要 3 条边，而每条边是两个面的边界，故 $2e \geq 3f$. 又 $f = 2 + e - n$, 故 $2e \geq 6 + 3e - 3n$, 即 $e \leq 3n - 6$.

若 G 是三角形无关的，即每个面至少有 4 条边，故 $2e \geq 4f$. 同理可得 $e \leq 2n - 4$.

例子 6. 前面已证明 κ_5 和 $\kappa_{3,3}$ 不是可平面图，下面由命题 5 来证明：对于 κ_5 ，有 $e = 10 > 9 = 3n - 6$ ；注意 $\kappa_{3,3}$ 是三角形无关的，且 $e = 9 > 8 = 2n - 4$ ，由命题 5 可知 κ_5 和 $\kappa_{3,3}$ 都不是可平面图.

下面我们将给出本节最重要的一个定理，它给出了可平面图的特征刻画，这就是著名的 Kuratowski 定理. 先看一个定义.

定义 7. 在图 G 的边上加入顶点得到的图 G' 称为图 G 的细分 (subdivision).

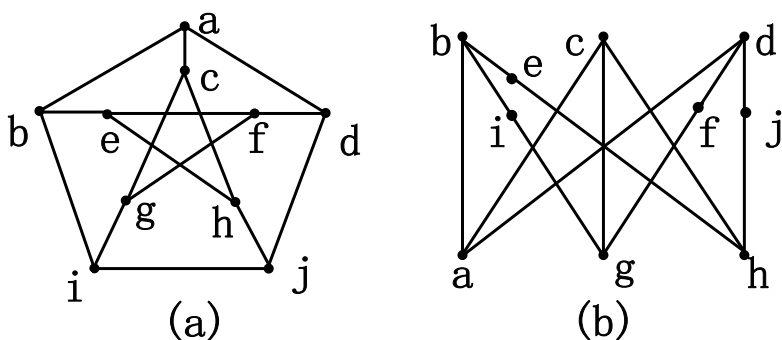


定理 10 (Kuratowski 1930). 一个图是可平面的当且仅当它不含 κ_5 或 $\kappa_{3,3}$ 的细分.

下面给出可平面图的一个特征.

定理 11 (Wagner 1937). 一个图是可平面的当且仅当它没有收缩到 κ_5 或 $\kappa_{3,3}$ 的子图.

例 12. 证明 Peterson 是不可平面的. 下图 (a) 是 Peterson 图，下图 (b) 是 Peterson 图的一个子图且是 $\kappa_{3,3}$ 的细分. 由定理 10 得，Peterson 图不是可平面的.



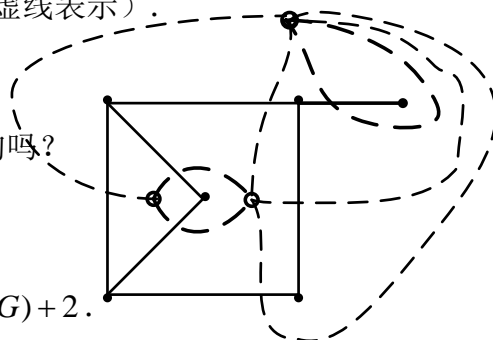
习题:

1. 用类似命题 3 中的方法证明 $K_{3,3}$ 不是可平面图.
2. 对顶点数 n , 面数 f 分别进行归纳, 证明定理 4.
3. 平面图 G 的对偶图 G^* (dual graph) 是一个平面图, 其顶点对应于 G 的面.

G^* 中两顶点间有一条边相连当且仅当这两顶点对应的 G 中的面有一个公共边界.

例如右下图给出一个平面图和其对偶图 (虚线表示).

- ① $(G^*)^*$ 与 G 一定同构吗?
- ② 可平面图的不同平面嵌入的对偶一定同构吗?
- ③ 如果 G 与 G^* 同构, 则称 G 是自同构的.



证明: 如果 G 是自同构的, 则 $2n(G) = e(G) + 2$.

举例说明对于一个平面图 H , 满足 $2n(H) = e(H) + 2$, 但 H 不是自同构的.

4. 证明: 对于简单平面图 G , 有 $\delta(G) \leq 5$.
5. ① 设 G 是连通平面图且 G 中最小圈得长度 (称为 G 的围长 (girth)), κ 不小于 3.

证明:
$$e(G) \leq \frac{\kappa(n(G)-2)}{\kappa-2}$$

②用①中结论证明 Peterson 图不是可平面的.

6. 证明: 正多面体只有 5 种 (提示: 首先说明每个正多面体对应一个平面图, 然后利用欧拉公式及度和公式得到关于不等关系式)

第二节 哈密尔顿图与欧拉图

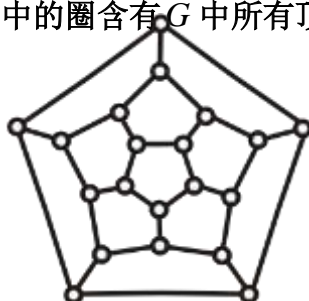
定义 1. 在图 G 中含有一切边的迹称为欧拉迹 (Euler 迹), 闭的欧拉迹称为欧拉回路 (Euler 回路). 若 G 中含有欧拉回路, 则称 G 为欧拉图.

下面给出欧拉图的判别法.

定理 1. 对于连通图 G , 下列条件等价:

- ① G 是欧拉图.
- ② $\forall v \in V(G), d(v)$ 是偶数.
- ③ G 可表示成无公共边的圈之并.

定理 2. 若图 G 中的圈含有 G 中所有顶点, 则称此图为哈密尔顿圈, 称 G 为哈密尔顿图.



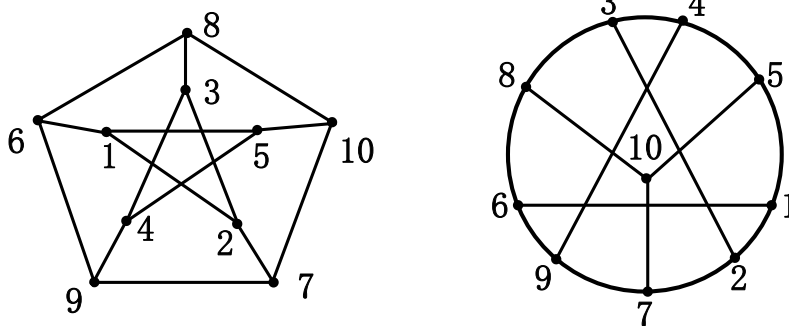
正十二面体

下面介绍一个哈密尔顿图的充分条件.

定理 3. 若 G 的阶为 $n \geq 3$, 且 $\forall v \in V(G), d(v) \geq \frac{n}{2}$, 则 G 是哈密尔顿图

注意: 此条件不是必要的.

下面说明 Peterson 图不是哈密尔顿图.



证明: (反证法) Peterson 图是 3-正则图, 故每个顶点均有一边不属于哈密尔顿圈 C .

不妨设 $12 \notin C$, 则 $16, 23, 27, 15 \in C$.

(1) 若 $8(10) \notin C \Rightarrow (10)7, (10)5 \in C$, $38, 86 \in C$. 此时已形成了一个圈, 矛盾.

(2) 若 $8(10) \in C$ 同样得出矛盾.

第三节 鸽巢原理与拉姆赛数

在组合数学中, 我们常常用鸽巢原理 (pigeonhole principle) 来证明某种排列或模式的存在.

定理 1. (鸽巢原理) 如果把 $k + 1$ 只鸽子放入 k 个鸽巢里, 那么至少有一个鸽巢包含两只以上的鸽子.

我们来研究一个比鸽巢原理更强的形式.

定理 2. 把 m 只鸽子放入 k 个鸽巢中, 至少有一个鸽巢里的鸽子数大于 $\left\lfloor \frac{m-1}{k} \right\rfloor$.

证明: 如果每个鸽巢中鸽子数至多是 $\left\lfloor \frac{m-1}{k} \right\rfloor$, 那么鸽子总数为

$$k \left\lfloor \frac{m-1}{k} \right\rfloor \leq m-1 < m$$

应用鸽巢原理的关键在于如何定义对象 (鸽子) 和类 (鸽巢), 下面举几个例子.

例 3. 如果 G 是一个 n 个顶点的简单图且满足 $\delta(G) \geq \frac{n-1}{2}$, 则 G 是连通的.

证明: $\forall u, v \in V(G)$, 如果 u, v 不相邻, 由于 $d(u) + d(v) \geq \frac{n-1}{2} + \frac{n-1}{2} = n-1$,

故至少有 $n-1$ 条边将 $\{u, v\}$ 与 $V(G) - \{u, v\}$ 相连. 又 $|V(G) - \{u, v\}| = n-2$, 由鸽

巢原理知，有一个顶点与上述 $n-1$ 条边中的两条边相关联。由于 G 是简单图，故这两条边分别与 u, v 相关联，即该顶点是 u 和 v 的公共相邻顶点，这样对于 G 中任意两个顶点，要么相邻，要么有公共的相邻顶点，故 G 连通。

例 4. (Erdős-Szekeres[1935]) 给定一个有 $n^2 + 1$ 个不同整数的序列，或者存在有 $n + 1$ 项的递增子序列或者存在有 $n + 1$ 项递减子序列。

证明：设给定的序列为 $a_1, a_2, a_3, \dots, a_{n^2+1}$ 。令 x_i 是开始于 a_i 的最长递增子序列的项数。如果存在 x_i 大于或等于 $n+1$ ，则定理得证。下设 $\forall i \in \{1, 2, \dots, n^2 + 1\}, 1 \leq x_i \leq n$ ， $n^2 + 1$ 个 x_i （鸽子）在数 1 到 n 之间（ n 个鸽巢），因此存在一个至少有

$$\left\lfloor \frac{n^2 + 1 - 1}{n} \right\rfloor + 1 = n + 1$$

只鸽子的鸽巢，即至少有 $n+1$ 个相等的 x_i 。下面证明与这 $n+1$ 个 x_i 对应的 a_i 形成一个递减子序列，从而完成定理证明。为此，只需要证明：如果 $i < j$ 且 $x_i = x_j$ ，则 $a_i > a_j$ 。如若不然，假设 $a_i \leq a_j$ ，即 $a_i < a_j$ （原序列由不同整数构成）。 a_i 和从 a_j 开始的最长递增子序列构成了一个长度为 $x_j + 1$ 的递增子序列，故 $x_i \geq x_j + 1$ ，与 $x_i = x_j$ 矛盾。

例 5. 6 个人中，能找到 3 个相互认识的人或 3 个相互不认识的人（假设两个人要么相互认识，要么相互不认识）。

证明：设 a 为 6 人中的一人，有鸽巢原理得，其余 5 个人中，和 a 相互认识的至少有 3 人或者和 a 相互不认识的至少有 3 人。不妨设 b, c, d 和 a 都相互认识。如果 b, c, d 中有两人相互认识，则这两个人和 a 就是相互认识的 3 个人。否则 b, c, d 就是 3 个相互不认识的人。

例 5 说明 6 个人中有 3 人相互认识或者有 3 人相互不认识。其实很容易看出若一群人中有 3 人相互认识或者有 3 人相互不认识，则这群人至少有 6 人。一个更一般的问题是：若一群人中有 k 个两两相互认识或者两两相互不认识，则这群人至少有多少人？这个答案记为 $R(k, l)$ ，称为 Ramsey 数 (Ramsey number)。由前面论述可知 $R(3, 3) = 6$ 。下面给出严格的定义：

定义 6. 设 $k, l \in \mathbb{Z}^+$ 且 $k, l \geq 2$ ，对于任意的 n 个元素的集合 S ，我们把 S 的

所有二元子集分成 X 和 Y 两类，如果下列条件之一成立：

1. 存在 S 的 k 元子集，其所有的二元子集都在 X 内；
2. 存在 S 的 l 元子集，其所有的二元子集都在 Y 内。

则称 n 有 (k, l) 拉姆塞性质 (Ramsey property). 具有 (k, l) 拉姆塞性质的最小数称为拉姆塞数，记为 $R(k, l)$.

下面的定理说明上述定义是合适的.

定理 7. 设 $k, l \in \mathbb{Z}^+$ 且 $k, l \geq 2$ ，则 $R(k, l)$ 存在.

第四节 容斥原理

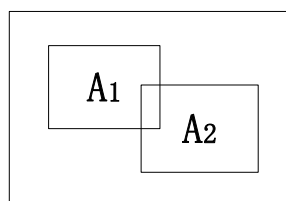
下面我们介绍一个基本的计数工具，这就是容斥原理.

设 A_1, A_2, \dots, A_n 都是有限集合，则一下公式成立.

$$\begin{aligned} & |A_1 \cup A_2 \cup \dots \cup A_n| \\ &= \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \sum_{1 \leq i < j < k \leq n} |A_i \cap A_j \cap A_k| - \dots + (-1)^{n-1} |A_1 \cap A_2 \cap \dots \cap A_n| \end{aligned}$$

证明：用归纳法证明.

当 $n = 2$ 时，即 $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$ ，显然成立（参照下图）



设 $n-1 (n \geq 3)$ 时公式成立，现证明当 n 时也成立.

$$\begin{aligned} & |A_1 \cup A_2 \cup \dots \cup A_n| \\ &= |A_1 \cup A_2 \cup \dots \cup A_{n-1}| + |A_n| - |(A_1 \cup A_2 \cup \dots \cup A_{n-1}) \cap A_n| \end{aligned}$$

由归纳假设得，

$$\begin{aligned}
& |A_1 \cup A_2 \cup \dots \cup A_n| \\
&= \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n-1} |A_i \cap A_j| + \dots + (-1)^{n-2} |A_1 \cap A_2 \cap \dots \cap A_{n-1}| \\
&\quad - \left[\sum_{i=1}^{n-1} |A_i \cap A_n| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j \cap A_n| + \dots + (-1)^{n-2} |A_1 \cap A_2 \cap \dots \cap A_{n-1} \cap A_n| \right] \\
&= \sum_{i=1}^n |A_i| - \sum_{1 \leq i < j \leq n} |A_i \cap A_j| + \dots + (-1)^{n-1} |A_1 \cap A_2 \cap \dots \cap A_n|
\end{aligned}$$

定义 1.2 设图 G 的顶点集为 $V(G) = \{v_1, v_2, \dots, v_n\}$, 边集为 $E(G) = \{e_1, e_2, \dots, e_m\}$. G 的邻接矩阵 (adjacency matrix) $A(G)$ 是一个 $n \times n$ 矩阵, 元素 $a_{i,j}$ 是以 v_i 和 v_j 为端点的边的数目. G 的关联矩阵 (incidence matrix) $M(G)$ 是一个 $n \times m$ 矩阵, 元素 $m_{i,j} = 1$, 当 v_i 是 e_j 的端点且 e_j 不是环. 否则, $m_{i,j} = 2$. 若 v_i 不是 e_j 的端点, 则 $m_{i,j} = 0$. 顶点 v 的度 (degree) 是其作为边的端点的个数, 记为 $d(v)$.

例 1.3 图 1 的邻接矩阵和关联矩阵分别如下:

$$\begin{array}{cc}
\begin{array}{c} v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5 \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{bmatrix} 1 & 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} &
\begin{array}{c} e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7 \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \\
A(G) & M(G)
\end{array}$$

注意: 邻接矩阵由顶点的顺序决定. 任意邻接矩阵都是对称的. 邻接矩阵法是将一个图储存于计算机的方法之一. 在关联矩阵中, 将某顶点对应的行的元素求和, 就得到该顶点度数.