1

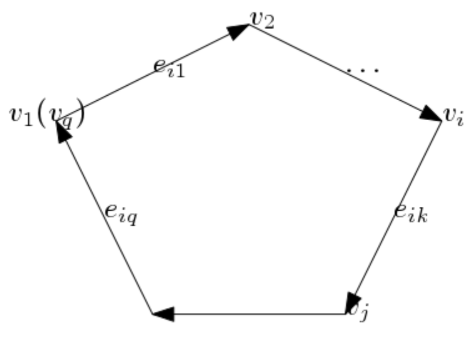
基本概念

2

道路与回路

2.1 道路与回路

定义 2.1.1. 有向图 G=(V,E) 中，若边序列 P=($e_{i1}, e_{i2}, \ldots, e_{iq}$)，其中 $e_{ik} = (v_l, v_j)$ 满足 v_l 是 e_{ik-1} 的终点，v_j 是 e_{ik+1} 的始点，就称 P 是 G 的一条**有向道路**。如果 e_{iq} 的终点也是 e_{i1} 的始点，则称 P 是 G 的一条**有向回路**。



一条有向的回路

例 2.1.1. 下图中，边序列（e_5, e_4, e_5, e_7 ）是有向道路，（e_5, e_4, e_5, e_7, e_3 ）是有向回路。（e_5, e_4, e_1, e_2 ）是简单有向道路，（e_5, e_4, e_1, e_2, e_3 ）是简单有向回路。（e_1, e_2 ）是初级有向道路，（e_1, e_2, e_3 ）是初级有向回路。

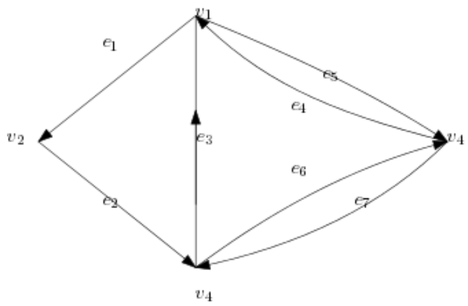


图 2.1

定义 2.1.2. 无向图 G=(V,E) 中，若点边交替序列 P=$(v_{i1}, e_{i1}, v_{i2}, e_{i2}, \ldots, e_{iq-1}, v_{iq})$ 满足 v_{ik}，v_{ik+1} 是 e_{ik} 的两个端点，则称 P 是 G 中的一条链，或道路。如果 $v_{iq} = v_{i1}$，则称 P 是 G 中的一个圈，或**回路**。
如果 P 中没有重复出现的边，称之为**简单道路**或**简单回路**，若其中结点也不重复，又称之为**初级道路**或**初级回路**。

思考非初级有向道路的简单有向道路有什么特征？

例 2.1.2. 下图中边序列 (e_4, e_5, e_4, e_6) 是道路，(e_4, e_5, e_4, e_6, e_3) 是回路；(e_4, e_5, e_1, e_2) 是简单道路，(e_4, e_5, e_1, e_2, e_3) 是简单回路；(e_1, e_2) 是初级道路，(e_1, e_2, e_3) 是初级回路。

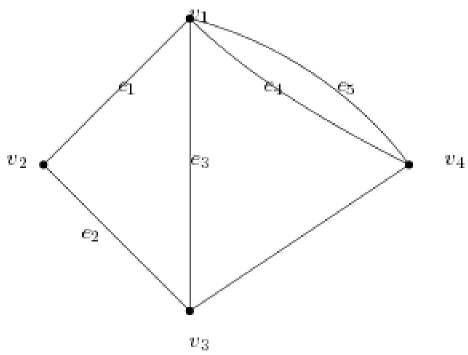


图 2.2

例 2.1.3. 设 C 是简单图 G 中含结点数大于 3 的一个初级回路，如果结点 v_i 和 v_j 在 C 中不相邻，而边 (v_i, v_j)∈E(G)，则称 (v_i, v_j) 是 C 的一条弦。若对每一个 $v_k V(G)$，都有 d(v_k)⩾3，则 G 中必含带弦的回路。

证明：在 G 中构造一条极长的初级道路 $P = (e_{i1}, e_{i2}, \ldots, e_{iq})$，不妨设 $e_{i1} = (v_0, v_1)$，$e_{il} = (v_{l-1}, v_l)$。由于 P 是极长的初级道路，所以 v_0 和 v_1 的邻接点都在该道路 P 上。由已知条件，d(v_0)⩾3，不妨

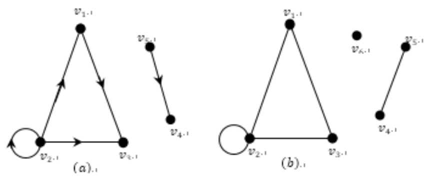


图 2.3

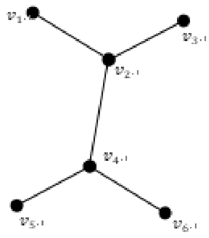


图 2.4

设 Γ(v_0)=$v_1, v_{ij}, v_{ik}, \ldots$。其中 1<j<k，这时 ($v_0, v_1, \ldots, v_i, v_0$) 是一条初级回路，而 (v_0, v_{ij}) 就是该回路中的一条弦。

例 2.1.4. 设 C 是简单图 G 中含结点数大于 3 的一个初级回路，如果结点 v_i 和 v_j 在 C 中不相邻，而边 (v_i, v_j)∈E(G)，则称 (v_i, v_j) 是 C 的一条弦。若对每一个 v_k∈V(G)，都有 d(v_k)⩾3，则 G 中必含带弦的回路。

证明:在 G 中构造一条极长的初级道路 P=($e_{i1}, e_{i2}, \ldots, e_{iq}$)，不妨设 e_{i1}=(v_0, v_1)，e_{il} = (v_{l-1}, v_l)。由于 P 是极长的初级道路，所以 v_0 和 v_1 的邻接点都在该道路 P 上。由已知条件，d(v_0)⩾3，不妨设 Γ(v_0)=$v_1, v_{ij}, v_{ik}, \ldots$。其中 1<j<k，这时 ($v_0, v_1, \ldots, v_{ik}, v_0$) 是一条初级回路，而 (v_0, v_{ij}) 就是该回路中的一条弦。

例 2.1.5. 设 G=(V,E) 是无向图，如果 V(G) 可以划分为子集 X 和 Y，使得对所有的 e=(u,v)∈E(G)，u 和 v 都分属于 X 和 Y，则称 G 是二分图。证明：如果二分图 G 中存在回路，则它们都是由偶数条边组成的。

证明： 设 C 是二分图 G 的任一回路，不妨设 v_0∈X 是 C 的始点，由于 G 是二分图，所以沿回路 C 必须经过偶数条边才能达到某结点 v_i∈X，因而只有经过偶数边才能回到 v_0

定义 2.1.3. 设 G 是无向图，若 G 的任意两结点之间都存在道路，就称 G 是**连通图**，否则称为**非连通图**。

如果 G 是有向图，不考虑其边的方向，即视之为**无向图**，若它是连通的，则称 G 是连通图。

若连通子图 H 不是 G 的任何连通子图的真子图，则称 H 是 G 的**极大连通子图**，或称**连通支**。显然 G 的每个连通支都是它的**导出子图**。

例 2.1.6. 图 2.1 和图 2.2 都是连通图，图 2.3 是非连通图。其中 (a) 有两个连通支，它们的结点集分别是v_1, v_2, v_3 和v_4, v_5 ，(b) 有三个连通支，其结点集是v_1, v_2, v_3 ，v_4, v_5 和v_6 。

例 2.1.7. 图 2.4 是连通图，它不含回路，而且在任意两结点之间都只有唯一的一条初级道路。这种图称为树，它是含边数最少的连通图。

例 2.1.8. 设 G 是简单图，证明当 m=1/2 (n-1)(n-2) 时，G 是连通图。

证明：假定 G 是非连通图，则至少含有 2 个连通支。设分别为 $G_1=(V_1,E_1)$，$G_2=(V_2,E_2)$。其中 $|V_1(G_1)|=n_1$，$|V_2(G_2)|=n_2$。$n_1+n_2=n$。由于 G 是简单图，因此

$$|E_1(G_1)| \leq 1/2n_1(n_1-1)$$

$$|E_2(G_2)| \leq 1/2n_2(n_2-1)$$

$$m \leq 1/2n_1(n_1-1)+1/2n_2(n_2-1)$$

由于 n_1≤n-1，n_2≤n-1，

所以

$$m \leq 1/2(n-1)(n_1-1+n_2-1)$$

$$=1/2(n-1)(n-2)$$

与已知条件矛盾，故 G 是连通图。

2.2 道路与回路的判定

通常可以利用邻接矩阵或搜索法判定某个图 G 的两结点间是否存在道路，或者判定它是否连通。首先介绍 **邻接矩阵**的判定方法。

设 $A=(a_{ij})_{n\times n}$ 是 G 的邻接矩阵。由 A 的定义，$a_{ij}=1$ 表示 $(v_i,v_j)\in E(G)$，即 v_i 可以通过某条边 e 到达 v_j，者说 G 中有道路从 v_i 到 v_j。根据矩阵乘法，设 $A^2=(a_{ij}^{(2)})$，有

$$a_{ij}^{(2)}=\sum_{k=1}^{n} a_{ik}.a_{kj}$$

$a_{ij}^{(2)}\neq 0$ 当且仅当存在 k，使 $a_{ik}=a_{kj}=1$。也就是说，如果 G 中存在结点 v_k，满足 (v_i,v_k)，$(v_i,v_k)\in E(G)$，即经过 2 条边 (v_i,v_k)，(v_k,v_j)，v_i 可以到达 v_j 时，$a_{ij}^{(2)}\neq 0$。同理，$A^l(ln)$ 中的元素 $a_{ij}^{(l)}\neq 0$ 表示了 v_i 可以经过 l 条边到达 v_j。因此令

$$P=A+A_2+\cdots+A_n,$$

如果 $p_{ij}=t$，说明 v_i 有 t 条道路可以到达 v_j。若 $p_{ij}=0$，即 n 步之内 v_i 不能到达 v_j，则在 G 中不存在 v_i 到 v_j 的路。否则，若 v_i 经过 l(l≤n) 步

可达 v_j，由**抽屉原理**，该道路上一定存在重复出现的结点 v_k，而 v_k 之间的这段路 C 是一个回路。删去这段回路 v_i 仍然可达 v_j。由于 G 中只存在 n 个不同的结点，所以只要 v_i 有道路到 v_j，一定有 $p_{ij}0$。

在许多实际问题中，往往只要求了解 v_i 与 v_j 之间是否存在道路。对此可以采用逻辑运算的方法，即

$$a_{ij}^{(l)} = \bigvee_{k=1}^{n} (a_{ik}^{(l-1)} \bigwedge a_{ij}), l = 2, 3, \cdots, n$$

相应地，

$$P = A \bigvee A^2 \bigvee \cdots \bigvee A^n$$

就是图 G 的道路矩阵。

用上述方法求 G 的道路矩阵，计算复杂性为 O(n^4)。以下介绍的 Warshall 算法是一个更好的方法，其计算复杂性是 O(n^3)。

Warshall **算法**

begin

1. $P \leftarrow A$,
2. for i=1 to n do
3.   for j=1 to n do
4.   for k=1 to n do

   $p_{jk} \leftarrow p_{jk} \bigvee (p_{jk} \bigwedge p_{ik})$。

例 2.2.1. 采用 Warshall 算法计算图 2.5 道路矩阵的过程是：

$$P \leftarrow \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$P(i=1) = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad P(i=2) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$P(i=3)=\begin{bmatrix}0&1&1&1&1\\0&0&1&1&1\\0&0&0&0&1\\0&0&0&0&0\\1&1&1&1&1\end{bmatrix}\quad P(i=4)=P(i=3)$$

$$P(i=5)=\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&1&1&1\\0&0&0&0&0\\1&1&1&1&1\end{bmatrix}$$

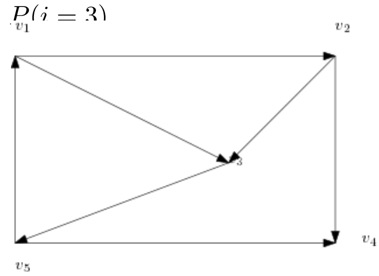


图 2.5

定理 2.2.1. Warshall 算法的结果是图 G 的道路矩阵。

证明：该定理的严格证明需要对三层循环分别使用归纳法。现只证其最外层循环。

基始：当 i=1 时，

$p_{jk}^{(l)}=p_{jk}\bigvee(p_{jl}\bigwedge p_{lk}),k=1,2,\cdots,n;j=1,2,\cdots,n$

$p_{jk}^{(1)}=1$ 当且仅当 $p_{jk}=1$ 或 $p_{j1}=p_{1k}=1$，其中 $p_{jk}=1$ 表明 v_j 直接可达 v_k，$p_{j1}=p_{1k}=1$ 表明 v_j 直接经过 v_1 可达 v_k。因此 $p_{jk}^{(l)}=1$ 当且仅当结点集v_j,v_1,v_k 之间有 v_j 到 v_k 的路。

i=2 时，$p_{jk}^{(2)}=p_{jk}^{(1)}(p_{j2}^{(1)}p_{2k}^{(1)})$，$k=1,2,\cdots,n$；$j=1,2,\cdots,n$。$p_{jk}^{(2)}=1$ 当且仅当 $p_{jk}^{(1)}=1$ 或 $p_{j2}^{(1)}=p_{2k}^{(1)}=1$，其中 $p_{jk}^{(1)}=1$ 表明结点集v_j,v_1,v_k 之间有 v_j 到 v_k 的道路；$p_{j2}^{(1)}$ 和 $p_{2k}^{(1)}$ 为 1 表明v_j,v_1,v_2,v_k 之间 v_j 有必通过 v_2 到达 v_k 的道路，因此，$p_{jk}^{(2)}=1$ 当且仅当结点集v_j,v_1,v_2,v_k 中有 v_j 到 v_k 的道路。

设 i=n-1 时，$p_{jk}^{(n-1)}=1$ 当且仅当结点集$v_j,v_1,v_2,\cdots,v_{n-1},v_k$ 之中有 v_j 到 v_k 的道路。

则 i=n 时，$p_{jk}^{(n)}=p_{jk}^{(n-1)}\bigvee(p_{jn}^{(n-1)}\bigwedge p_{nk}^{(n-1)})k=1,2,\cdots,nj=1,2,\cdots,n$。由归纳假设，$p_{jk}^{(n-1)}=1$ 表明结点集$v_j,v_1,v_2,\cdots,v_{n-1},v_k$ 中有 v_j 到 v_k 的路，$p_{jn}^{(n-1)}=p_{nk}^{(n-1)}=1$ 表明结点集$v_j,v_1,v_2,\cdots,v_{n-1},v_k$ 中 v_j 有通过 v_n 到达 v_k 的道路。因此，$p_{jk}^{(n)}=1$ 即是结点集$v_j,v_1,\cdots,v_n,v_k$ 之中有 v_j 到 v_k 的道路。

采用搜索的方法判断 G 中某一结点 v_0 到另一结点 v_j 是否存在道路经常更加方便。常用的搜索法有广探法 (Breadth First Search) 和深探法 (Depth First Search)。

广探法 (BFS)是从 G 的任一结点 v_0 开始，找它的直接后继集 $^+(v_0)$，记为 A_1，然后对 A_1 中的每一结点分别找它们的直接后继集，这些直接后继集的并记为 A_2。依此类推，直至达到目的。为了避免结点的重复搜索，可以首

先对全部结点都给一个标记“0”，当 v_i 被搜索到时，如果其标记为 0，则 v_i 进入直接后继集，同时标记改为 1，否则由于 v_i 已被搜索因此不再进入直接后继集。

例 2.2.2. 用 BFS 方法找图 2.6 中 v_1 到 v_4 的一条道路。**解：**如果采用正

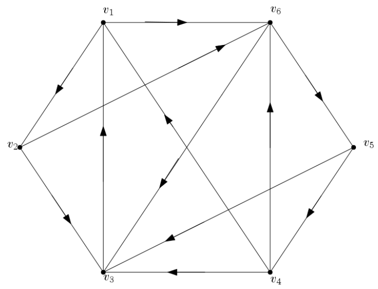
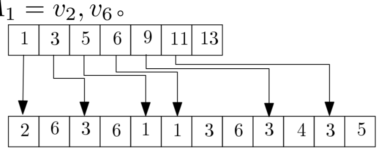


图 2.6

向表的输入结构，则有

$\because \Gamma^+(v_1) = v_2, v_6, \therefore A_1 = v_2, v_6$。



$\because \Gamma^+(v_2) = v_3, v_6,$

$\Gamma^+(v_6) = v_3, v_5, \therefore A_2 = v_3, v_5$。

$\because \Gamma^+(v_3) = v_1, \Gamma^+(v_5) = v_3, v_4,$

$\therefore A_3 = v_4$

从上例中可知，用 BFS 方法求两点间道路的计算复杂性是 O(m)。

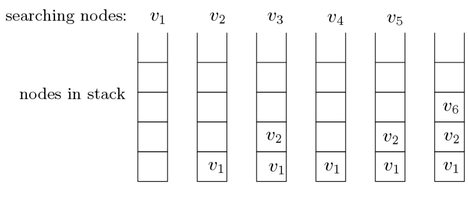


图 2.7

深探法 (DFS)的特点与 BFS 截然不同。它从某一结点 v_0 开始，只查找 v_0 的某个直接后继 v_1，记下 v_1 的父亲 v_0，然后再找 v_1 的某个未搜索过的直接后继 v_2。依此类推。当从某个结点 v_j 无法再向下搜索时，退回到它的父亲 v_{j-1}，然后再找 v_{j-1} 的另一个未查过的直接后继。形象地说，DFS 的特点是尽量向下搜索，只有碰壁才回头。

采用栈结构以及前述的标记结点的方法可以完成 DFS 的搜索过程。

例 2.2.3. 用 DFS 方法找图 2.6 中 v_1 到 v_4 的一条道路。

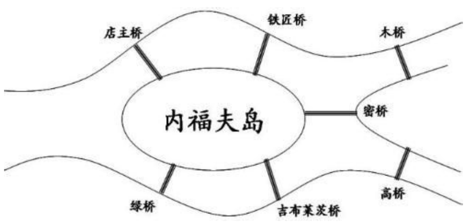
解：数据输入依然采用正向表。v_1 的第一个直接后继是 $v_2 v_1$ 进栈；v_2 的第一

个后继是 v_3v_2 进栈。v_3 的后继是 v_1，但已标记，故退栈。v_2 的另一个后继是 v_6，v_2 进栈；v_6 的第 1 个后继是已标记结点 v_3，第 2 个后继是 v_5，v_6 进栈。v_5 的后继是 v_4。至此，已搜索到 v_1 到 v_4 的一条道路。整个搜索过程可用图 2.7 形象地表示。其计算复杂性也是 O(m)。

2.3 欧拉道路与回路

2.3.1 欧拉道路的引入

1736 年瑞士著名数学家欧拉 (Leonhard Euler) 发表了图论的第一篇论文“哥尼斯堡七桥问题”。这个问题是这样的：哥尼斯堡城被 Pregel 河分成了 4 部分，它们之间有 7 座桥。如图 2.8 所示。当时人们提出了一个问题，能否从城市的某处出发，过每座桥一次且仅一次最后回到原处。欧拉的文章漂亮地解决了这个问题。他把 4 块陆地设想为 4 个结点，分别用 A、B、C、D 表示，而将桥画成相应的边，如图 2.9。于是问题转化为在该图中是否存在经过每条边一次且仅一次的回路。欧拉的论文给出了解决这类问题的准则，并对七桥问题给出了否定的结论。



哥尼斯堡七桥

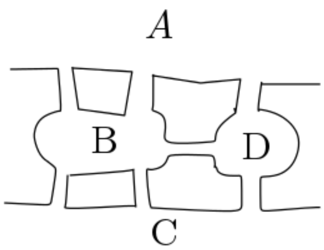


图 2.8

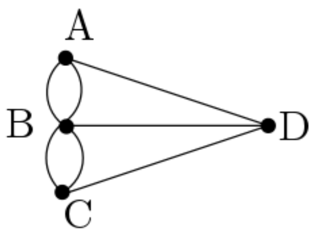


图 2.9

定义 2.3.1. 给定无孤立结点的无向图 G, 经过图 G 每边一次且仅一次的迹称为**欧拉路**。无向连通图 G=(V,E) 中的一条经过所有边的简单回路 (道路) 称为 G 的**欧拉回路 (道路)**。

定理 2.3.1. 无向连通图 G 存在欧拉回路的充要条件是 G 中各结点的度都是偶数。

证明：必要性。若 G 中有欧拉回路 C，则 C 过每一条边一次且仅一次。对任一结点 v 来说，如果 C 经由 e_i 进入 v，则一定通过另一条边 e_j 从 v 离开。因此结点 v 的度是偶数。
充分性。由于 G 是有穷图，因此可以断定，从 G 的任一结点 v_0 出发一定存在 G 的一条简单回路 C。这是因为各结点的度都是偶数，所以这条简单道路不可能停留在 v_0 以外的某个结点，而不能再向前伸延以至构成回路 C。如果 E(G)=C，则 C 就是欧拉回路，充分性得证。否则在 G 中删去 C 的各边，得到 $G_1 = G - C$。G_1 可能是非连通图，但每个结点的度保持为偶数。这时，G_1 中一定存在某个度非零的结点 v_i，同时 v_i 也是 C 中的结点。否则 C 的结点与 G_1 的结点之间无边相连，与 G 是连通图矛盾。同样理由，从 v_i 出发，G_1 中 v_i 所在的连通支内存在一条简单回路 C_1。显然 $C \bigcup C_1$ 仍然是 G 的一条简单回路，但它包括的边数比 C 多。继续以上构造方法，最终有简单回路 $C' = C \bigcup C_1 \bigcup \cdots \bigcup C_k$，它包含了 G 的全部边，即 C’ 是 G 的一条欧拉回路。
以上采用了构造性证明的方法，即证明过程本身就给出了问题求解的步骤。

例 2.3.1. 试找出图 2.10 的一条欧拉回路。

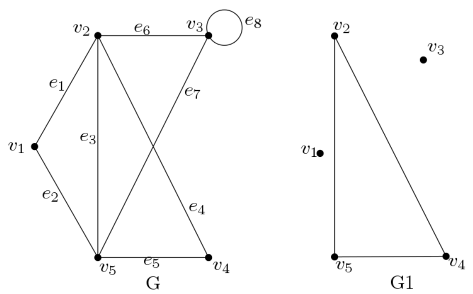


图 2.10

解：从任一点，比如 v_1 开始，可构造简单回路 $C = (e_1, e_6, e_8, e_7, e_2)$。$G_1 = G - C$ 中的 $v_2 v_5$ 度非零且是 C 中的结点，从 v_2 开始 G_1 中有简单回路 $C1 = (e_3, e_5, e_4)$。因此 $C \bigcup C_1 = (e_1, e_3, e_5, e_4, e_6, e_8, e_7, e_2)$ 包含了 G 的所有边，即是 G 的一条欧拉回路。

推论 2.3.1. 无向连通图 G 中只有 2 个度为奇的结点，则 G 存在欧拉道路。
证明：设 v_i 和 v_j 是两个度为奇数的结点。作 $G' = G + (v_i, v_j)$，则 G’ 中各点的度都是偶数。由定理 2.3.1，G’ 有欧拉回路，它包含边 (v_i, v_j)，删去该边，得到一条从 v_i 到 v_j 的简单道路，它恰好经过了 G 的所有边，亦即是一条欧拉道路。

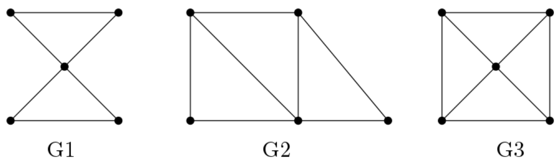
推论 2.3.2. 若有向连通图 G 中各结点的正、负度相等，则 G 存在有向欧拉回路。

其证明与定理 2.3.1 的证明相仿。

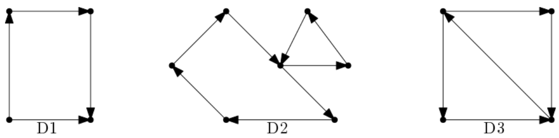
例 2.3.2. 七桥问题中既不存在欧拉回路也不存在欧拉道路。

例 2.3.3. 设连通图 G=(V,E) 有 k 个度为奇数的结点，证明 E(G) 可以划分成 k/2 条简单道路。
证明:由性质 1.1.2，k 是偶数。在这 k 个结点间增添 k/2 条边，使每个结点都与其中一条边关联，得到 G’，G’ 中各结点的度都为偶数。由定理 2.3.1，G’ 中有欧拉回路 C，这 k/2 条边都在 C 上且不相邻接。删去这些边，得到 k/2 条简单道路，它们包含了 G 的所有边。亦即 E(G) 划分成了 k/2 条简单道路。

例 2.3.4. 下图中，图 G1 是欧拉图；图 G2 不是欧拉图，但 G2 中存在欧拉路；图 G3 中即不存在欧拉回路，也不存在欧拉路。



例 2.3.5. 下图中，图 D1 中既不存在有向欧拉回路，也不存在有向欧拉路；图 D2 是欧拉图；图 D3 不是欧拉图，但 D3 中存在有向欧拉路。



例 2.3.6. 蚂蚁比赛问题：甲，乙两只蚂蚁分别位于 2.11 图中的节点 a,b 处，并设图中的边长度是相等的。甲乙进行比赛：从它们所在的结点出发，走过图中的所有边，最后到达结点 c 处。如果它们的速度相同，问谁先到达目的地？
解：在图中仅有两个度数为奇数的结点 b,c 因而存在从 b 到 c 的欧拉通路，蚂蚁乙走到 c 只要一条欧拉通路，边数为 9 条。而蚂蚁甲要走完所有的边到达 c，至少要先走一条边到达 b，再走一条欧拉通路，因而它至少要走 10 条边才能到达 c，所以乙必胜。

例 2.3.7. 一个编码盘分成 16 个相等的扇面，每个扇面分别由绝缘体和导体组成，可表示 0 和 1 两种状态，其中 a，b，c，d 四个位置的扇面组成一组

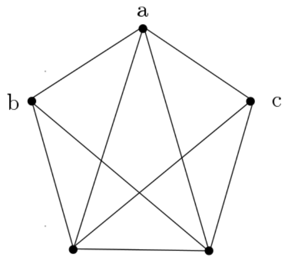


图 2.11

二进制输出，如图 2.12 所示。试问这 16 个二进制数的序列应如何排列，才恰好能组成 0000 到 1111 的 16 组四位二进制输出，同时旋转一周后又返回到 0000 状态？

解：我们发现如果从状态 $a_1a_2a_3a_4(a_i = 01)$ 逆时针方向旋转一个扇面，那

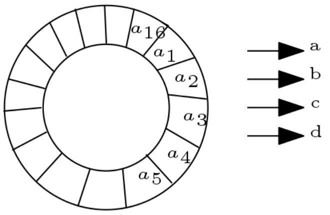


图 2.12

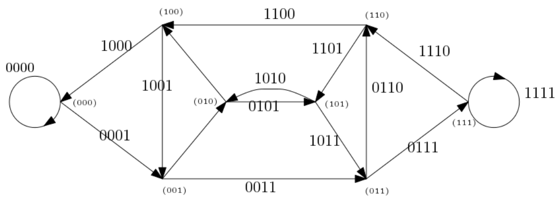


图 2.13

么新的输出是 $a_2a_3a_4a_5$，其中有三位数字不变。因此可以用 8 个结点表示从 000 到 111 这 8 个二进制数。这样从结点 ($a_{i-1}a_ia_{i+1}$) 可以到达结点 ($a_ia_{i+1}0$) 或 ($a_ia_{i+1}1$)，其输出分别为 ($a_{i-1}a_ia_{i+1}0$) 和 ($a_{i-1}a_ia_{i+1}1$)，这样可以得到图 2.13。它是有向连通图，共有 16 条边，且每结点的正、负度相等。由推论 2.3.2，它存在有向欧拉回路。其中任一条都是原问题的解，比如 (0000101001101111) 就是一种方案。

2.3.2 科学家的故事 -莱昂哈德 . 欧拉

莱昂哈德 . 欧拉

莱昂哈德 ．欧拉（Leonhard Euler ，1707 年 4 月 15 日～ 1783 年 9 月 18 日），瑞士数学家、自然科学家。1707 年 4 月 15 日出生于瑞士的巴塞尔，1783 年 9 月 18 日于俄国圣彼得堡去世。欧拉出生于牧师家庭，自幼受父亲的影响。13 岁时入读巴塞尔大学，15 岁大学毕业，16 岁获得硕士学位。

欧拉是 18 世纪数学界最杰出的人物之一，他不但为数学界作出贡献，更把整个数学推至物理的领域。他是数学史上最多产的数学家，平均每年写出八百多页的论文，还写了大量的力学、分析学、几何学、变分法等的课本，《无穷小分析引论》、《微分学原理》、《积分学原理》等都成为数学界中的经典著作。在几何方面，欧拉解决了哥尼斯堡七桥问题，成为图论、拓扑学的滥觞。欧拉对数学的研究如此之广泛，因此在许多数学的分支中也可经常见到以他的名字命名的重要常数、公式和定理。

此外欧拉还涉及建筑学、弹道学、航海学等领域。

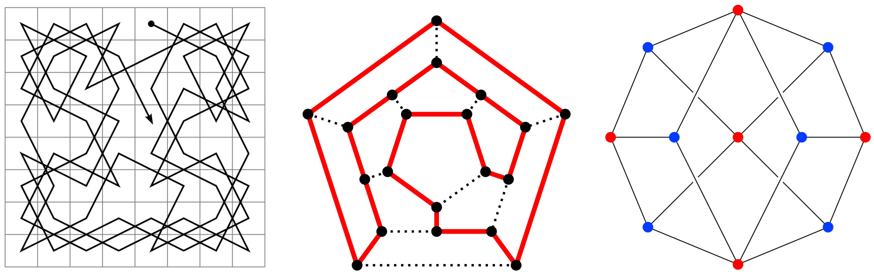
2013 年 4 月 15 日是欧拉诞辰的 306 周年，谷歌更换了首页涂鸦向这

位数学天才致敬。在那天的谷歌涂鸦中，融入了许多欧拉的数学成就。

2.4 哈密尔顿回路

2.4.1 哈密尔顿回路的引入

一个包含一个无向图中所有的点的初级回路被称作**哈密尔顿回路** (Hamilton Cycle)。这源于 1857 年 Sir William Hamilton 发明的一种游戏——遍历一个正十二面体，不能经过一个点两次。一个含有哈密尔顿回路的图称作**哈密尔顿图** (Hamiltonian)。事实上，在哈密尔顿之前，1759 年，欧拉就已经研究了在一个国际象棋棋盘上骑士的遍历问题 (Knight’s Tour on a Chess Board)(图 a 给出了一个解)。如果我们对旅行商问题再加上一重限制，两个城市之间的旅行费用只有 1 和 ∞ （也就是说不可能经过这条边)，那么这个 TSP 问题就变成了这个图中所有的旅行费用为 1 的边中是否存在一条哈密尔顿回路。然而，直到现在，即使这种 TSP 问题的特殊情况仍然没有解决：没有有效算法构造图中的哈密尔顿回路，虽然是否真的有这样的算法也不知道。



(a) 骑士遍历问题的一个解　(b) 正十二面体的一个哈密尔顿回路　(c) 赫歇尔图

图 2.14: 正十二面体图（图 b）是一个哈密尔顿图，而赫歇尔图（图 c）则不是

但是，边不交哈密尔顿回路问题现在我们已经可以很容易解决，这个问题在如下的例子中给出。

例 2.4.1. 约翰得到了 n(≥ 2) 种宝石，他想把这 n 种宝石串成几条 n-长的项链（每条项链中都含有每种宝石一颗），他希望自己串成的每条项链都本质不同，请问他最多能串出几条项链。两条项链**本质不同**，当且仅当每种宝石相邻的宝石种类都不同。

解 我们第一步估算上界：将 n 种宝石记作 $v_0, v_1, v_2, \ldots, v_{n-1}$，作完全图 K_n，则此题转化为计算完全图 K_n 中边不交 H 回路计数问题；完全图中一共有 $C_n^2 = \frac{n(n-1)}{2}$ 条边，每条 H 回路长为 n，所以至多存在 $[\frac{C_n^2}{n}] = [\frac{n-1}{2}]$ 条边不交 H 回路。
第二步可以构造一个解。
$n = 2k+1$ 时，如下图 (Figure 2) 将 $v_1, v_2, \ldots, v_{2k}$，排列在一个圆上，将 v_0 放在圆心。联结 $(v_0, v_1, v_2, v_{2k}, v_3, v_{2k-1}, \ldots, v_{k-1}, v_{k+3}, v_k, v_{k+2}, v_{k+1}, v_0)$ 形成一条 H 回路；删除联结线，下面将 $v_1, v_2, \ldots, v_{2k}$ 命名为顺时针下一个点的名字，也即，将 v_{2k-1} 命名为 v_{2k}，v_{2k} 命名为 v_1，v_1 命名为 v_2，v_2 命名为 v_3；重复执行联结操作，这样就得到 k 条边不交 H 回路。
$n = 2k+2$ 时，$2 \mid k$ 时，将 v_{2k+1} 插入 $v_{\frac{k+2}{2}}$ 和 $v_{\frac{3k+2}{2}}$ 的边中；$2 \nmid k$ 时，将 v_{2k+1} 插入 $v_{\frac{k+3}{2}}$ 和 $v_{\frac{3k+3}{2}}$ 的边中；依次就可以构造出 k 条边不交 H 回路。
综上，约翰可以串出 $[\frac{n-1}{2}]$ 条本质不同的项链。

定理 2.4.1. $n \geq 2$ 的完全图 K_n 可以被分解成边不交哈密尔顿回路。

Proof. 直接参照上例即可。 □

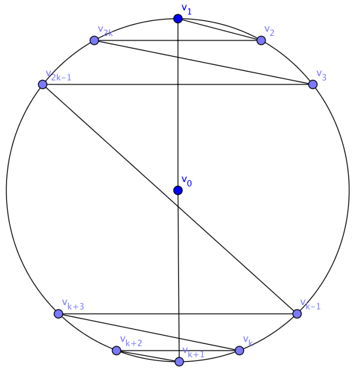


图 2.15：一种构造方案

2.5 哈密尔顿回路的几个重要判定定理

定理 2.5.1. 对于阶大于 3 的连通图 G，能够满足

$$\forall x, y \in G \wedge (x, y) \notin G \Rightarrow d(x) + d(y) \geq k$$

如果 $k = n$ 那么 G 就是一个哈密尔顿图，而如果 $k < n$ 那么 G 中就含有一条 k-长路，以及一个长度至少为 $\frac{k+2}{2}$ 的回路。

Proof. 假设 G 不是哈密尔顿图，我们找到 G 中的最长道路 P$(= x_1 x_2 \dots x_l)$。由于 P 是最长道路，所以 P 是极长道路。考虑

$$\Gamma(x_1) = \{x_j | (x_1, x_j) \in G\}, \Gamma^+(x_l) = \{x_{j+1} | (x_j, x_l) \in G\}.$$

我们可以断言这两个集合是不交的。否则就会产生回路，进而与 G 的连通性和非哈密尔顿图的性质违，这点留给读者自己证明。那么由定理中的不等式，我们有

$$k \leq d(x_1) + d(x_l) = \#\Gamma(x_1) + \#\Gamma^+(x_2) \leq l - 1 \leq n - 1.$$

$\#S$ 表示集合 S 的大小。如果 $k = n$，现在就已经产生了矛盾，G 是一个哈密尔顿图。如果 $k < n$，那么 G 中就存在一条长度为 $l - 1 = k$ 的路。如果再考虑 x_1 和 x_l 度的关系，不妨 $d(x_l) > d(x_1)$，也即 $d(x_l) \geq \frac{k}{2}$，我们就能够找到一个长度至少为 $\frac{k+2}{2}$ 的回路了。 □

定义 2.5.1. 若 v_i 和 v_j 是简单图 G 的不相邻结点，且满足 $d(v_i) + d(v_j) \geq k$，那么在 G 中增加边 (v_i, v_j)，重复这个过程，直到不再有这样的结点对为止。最终得到的图称为 G 的 k-闭包，记作 $C_k(G)$。

推论 2.5.1. 如果 $\delta(G) \geq \frac{n}{2}$，那么图 G 是哈密尔顿图。

定理 2.5.2. 图 G 是哈密尔顿图，当且仅当 $C_n(G)$ 是；图 G 有哈密尔顿路，当且仅当 $C_{n-1}(G)$ 有。

Proof. 这是**定理** 0.2.1的简单推论。 □

下面介绍一个中国数学家范更华给出的一个充分性判定条件。

定理 2.5.3 (范更华). 对于一个 2-连通图，如果对于任意一对距离为 2 的结点 $x, y, d(x, y) = 2$，都有

$$max\{d(x), d(y)\} \geq \frac{n}{2},$$

那么 G 是哈密尔顿图。

Proof. 略 □

再介绍一个非常实用的平面图具有哈密尔顿圈的必要条件。(尽管我们还没有严谨地定义过平面图)

定理 2.5.4 (Kozyrev and Grinberg). 如果一个平面图含有哈密尔顿圈 C，用 f_k, g_k 表示 C 内部和外部的 k 边形的数量，我们有

$$\sum_{k\geq 3}(k-2)(f_k - g_k) = 0.$$

这个定理可以很方便地证明一类平面图的非哈密尔顿性。

例 2.5.1 (Grinberg 图). Figure 3 不含哈密尔顿回路。

Proof. Figure 3 中只有五边形、八边形和九边形。

$$3(f_5 - g_5) + 6(f_8 - g_8) + 7(f_9 - g_9) = 0.$$

所以，

$$f_9 \equiv g_9(mod3)$$

而 $f_9 + g_9 = 1$，所以不含哈密尔顿回路。 □

例 2.5.2 (Grinberg 图). Figure 3 不含哈密尔顿回路。

Proof. Figure 3 中只有五边形、八边形和九边形。

$$3(f_5 - g_5) + 6(f_8 - g_8) + 7(f_9 - g_9) = 0.$$

所以，

$$f_9 \equiv g_9(mod3)$$

而 $f_9 + g_9 = 1$，所以不含哈密尔顿回路。 □

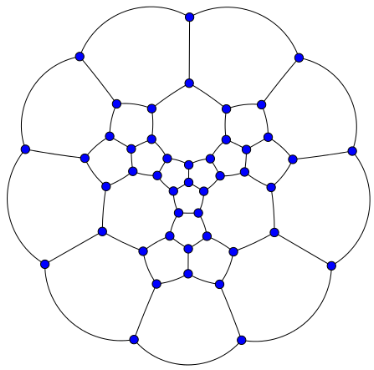


图 2.16: Grinberg 图

2.6 坚韧度与哈密尔顿性

定理 2.6.1. 如果图 G=(V,E) 是哈密尔顿图，那么

$$\forall S \subset V \Rightarrow \sigma(G-S) \leq \#S,$$

这里 $\sigma(G-S)$ 表示 $G-S$ 的分支数。

Proof. 找到哈密尔顿回路 C，构造图 $G' = (V', E')$，s.t.

$$V' = V \cap C, E' = E.$$

那么新图 G' 只包含这一个圈，如果去除掉 $\#S$ 个点剩下就有最多 $\#S$ 个分支，而原图的边更多一些，

$$\sigma(G-S) \leq \sigma(G'-S) \leq \#S$$

□

这个定理很容易证明，然而由这个定理产生的对于哈密尔顿图充分条件的猜想，却很有意思。

定义 2.6.1 (Kozyrev and Grinberg). $t = min\frac{\#S}{\sigma(G-S)}$，则称 G 是 t-**坚韧图**。

上面的定理表明：哈密尔顿图一定是 1-**坚韧图**。

Chvatal 认为图的坚韧性和哈密尔顿性应当存在双向的判定关系。提出了如下的猜想。

猜想 2.6.1. 存在 t 满足任何 t-坚韧图都是哈密尔顿图。

他给出了 $\frac{3}{2}$-坚韧非哈密尔顿图。所以推测 t 应当等于 2。因为这样的话就和 Fleischner’s theorem 一致。

定理 2.6.2 (Fleischner). 如果 G 是一个 2-点连通图，那么 G^2 是哈密尔顿图。其中 G^2 中两点存在边当且仅当两点在 G 中距离小于等于 2。

之后，Thomassen 发现了 $t > \frac{3}{2}$ 的非哈密尔顿图，Enomoto 等人发现了 $(2-\epsilon)$-坚韧图对任意 $\epsilon > 0$ 没有 2- 因子。

定义 2.6.2. 一个 k-因子是图的一个生成 k-正则子图。

Enomoto 的这个结论说明作为哈密尔顿图的判定依据的坚韧度至少为 2。如果 Chvatal 的猜想成立，那么将证明两个开放了二十余年的猜想。

猜想 2.6.2. 任意 4-连通的点边对偶图是哈密尔顿图。

猜想 2.6.3. 任意 4-连通的不含 $K_{1,3}$ 子图的图是哈密尔顿图。

近些年这两个猜想被证明是等价的。然而，人们却发现并不是每一个 2-坚韧图都是哈密尔顿图。事实上，我们有如下的定理。

定理 2.6.3 (D. Bauer et al). $\forall \epsilon > 0$，存在 $(\frac{9}{4} - \epsilon)$- 坚韧的非哈密尔顿图。

所以，关于作为哈密尔顿图的充分条件的坚韧度是否存在还是一个开放的问题。

2.7 旅行商问题

2.7.1 旅行商问题

上节讨论的哈密顿回路不涉及边的长度（权值）。但是在许多实际问题中，每条边都可以有自己的长度（权值）。如有若干个城市，任何城市之间的距离都是确定的，旅行商从某城市出发，必须经过每一个城市且只经过一次，最后回到出发城市。问如何事先确定好一条最短的路线，使其旅行的距离最短。这个问题便是著名的**旅行商问题** (Traveling Salesman Problem)，又记作 TSP **问题**。在 19 世纪，旅行商问题被数学形式化为给定一个正权完全图，求其总长最短的哈密顿回路。

即使是最朴素形式的旅行商问题也在很多领域有应用，如规划、物流、微电子元件的制造乃至 DNA 序列上，在这些应用中，城市的概念被替换为顾客、销售点、DNA 片段等，距离的概念被替换为旅行时间、代价或者 DNA 片段之间的相似性。

但是容易知道，n 个节点的完全图存在 1/2 (n-1)! 个不同的 H 回路。如果使用穷举搜索法对 TSP 问题进行求解，即使使用每秒能进行十亿亿次浮点数运算的“神威太湖之光”超级计算机，对于 30 个点的情况也需要 1.4×10^4 个世纪。事实上在 20 世纪 70 年代旅行商问题就已被证明是 NP-complete 问题。

在 1948 年美国兰德 (Rand) 公司向推动旅行商问题解决者颁发奖励后，旅行商问题成为近代组合优化领域的一个著名问题。而在当时兰德公司的三位专家通过手工和计算机相结合的办法，创造了周游 49 个城市的纪录。

目前对于旅行商问题的求解方法主要分为两类：精确算法与近似算法。

其中精确算法主要包括穷举法 (复杂度为 O(n!))，和动态规划法 (复杂度为 O($n^2 2^n$)) 等。通过对于穷举法加入剪枝技巧我们还可以得到分支与界法 (branch-and-bound method，又称作分支限界法)。在上个世纪 60 年代，理查德 · 卡普 (Richard Manning Karp) 通过分支与界法把旅行商问题的纪录提高到了 65 个城市。而目前确定性算法的最高记录是 2006 年的 85900 个城市，其使用的方法是 branch-and-bound 和 branch-and-cut 算法。

在本节中我们将介绍一个确定性算法**分支与界法**和一个近似算法**便宜算法（又称最近邻算法）**。

2.7.2 分支与界法

分支与界法（branch-and-bound method，**又称作分支限界法）**是解决旅行商问题的一个确定性算法。算法基本思想是对边按权值排序后，按顺序搜索所有可能成为解的 H 回路，并通过已有解的上界对搜索树进行剪枝。其算法伪代码如下：

算法 1 Huffman 算法

1: 对 t($t \geq 2$) 个权值进行排序，使得

$$w_{i1} \leqslant w_{i2} \leqslant ... \leqslant w_{it}$$

2: 计算 $w_i = w_{i1} + w_{i2}$ 作为中间结点 v_i 的权，v_i 的左儿子是 v_{i1}，右儿子是 v_{i2}。在权序列中删去 w_{i1}, w_{i2}，加入 w_i，$t \leftarrow t-1$。若 t=1，结束。否则转（1）。