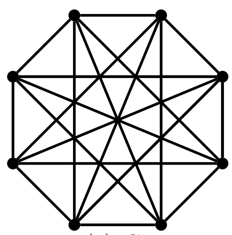
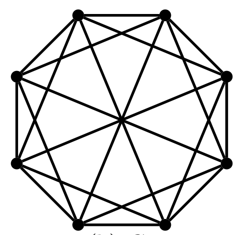
武汉大学计算机学院
《离散数学》第七次练习答案

§1 设G, H是简单无向图, 试证明$G \cong H$当且仅当$\overline{G} \cong \overline{H}$.

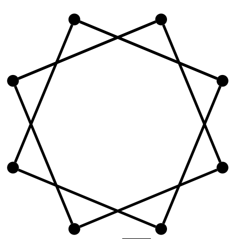
证明: 设$G \cong H$, 且G, H是简单图，则存在双射$\Phi : V(G) \to V(H)$, 且G和H边与结点的关系在Φ下保持，即若$\{u, v\} \in E(G)$ iff $\{\Phi(u), \Phi(v)\} \in E(H)$. 这样$\{u, v\} \notin E(G)$ iff $\{\Phi(u), \Phi(v)\} \notin E(H)$, 即 $\{u, v\} \in E(\overline{G})$ iff $\{\Phi(u), \Phi(v)\} \in E(\overline{H})$. 故 $\overline{G}$ 和 $\overline{H}$ 边与结点的关系在 Ψ 下也保持. 因此 $\overline{G} \cong \overline{H}$. 同理可证若$\overline{G} \cong \overline{H}$, 则 $G \cong H$. □

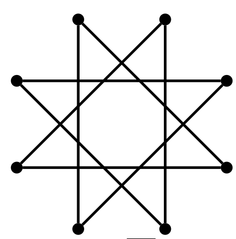
§2 利用上题的结果证明下述两图不同构：


(a) G_1


(b) G_2

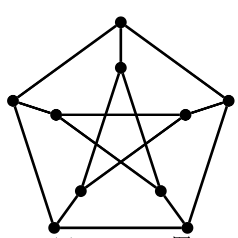
证明

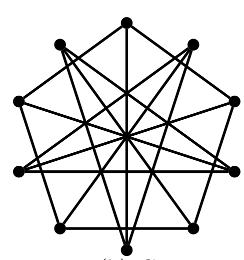

(a) $\overline{G_1}$


(b) $\overline{G_2}$

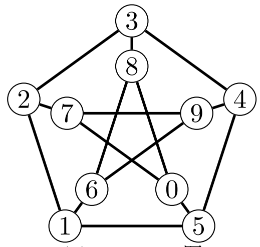
$\overline{G_1}$是由两个C_4连通分支组成的非连通图，而$\overline{G_2}$是C_8. 因此$\overline{G_1}$与$\overline{G_2}$不同构.

§3 试证明下述两图同构：

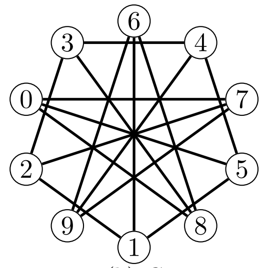

(a) Petersen图


(b) G

证明



(a) Petersen图

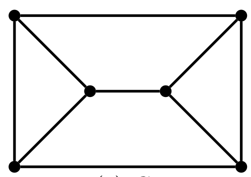


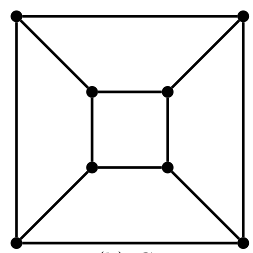
(b) G

§4 试用图示的方式说明：

(1) G_1可分解3条没有共同边的且长度均为3的基本路径；

(2) G_2可分解4个没有共同边的星$K_{1,3}$, 也可分解为4条没有共同边的且长度均为3的基本路径.

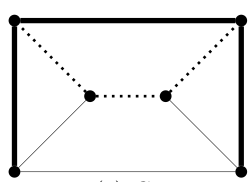

(a) G_1

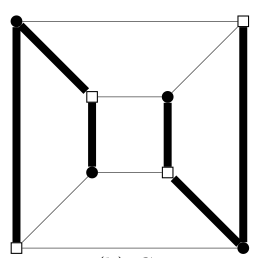

(b) G_2

解

(1) 分解的三条路径如下图粗线，细线和点线边所示;

(2) 分解后的$K_{1,3}$的中心如图中正方形结点，而4条路如粗线和细线所示.


(a) G_1


(b) G_2

§5 试用反正法证明Petersen图没有长度为7的基本回路.
证明 设Petersen图有一个长度为7的基本回路C. 由于Petersen图的每个结点的度数均等于3, 因此C的每个点还可引出一条边. 若$V(C)$中存在一点v其所引的第3边的另一端点也在C上，则

必导致Petersen图有一个长度不大于4的基本回路, 这与Petersen图的围长是5矛盾. 因此$V(C)$上所有点所引的第3边一定不在C上，即剩余的3个点将承载由$V(C)$中7 个点所引出的第3边. 根据鸽巢原理，存在$x \notin V(C)$且x所引的3边均落在C上. 再次鸽巢原理，x所引的3边的在$V(C)$上的端点一定有两个或在C中相邻，或在C中仅间隔一个点(间隔两个点需要圈上至少9个点). 这样可用x所引的到C上间隔小于2的两条边加上C的边构造一个长度不大于4的基本回路，这与围长5矛盾.

§6 试证明围长不小于4的k正则图G至少有$2k$个结点. (提示：考虑边xy的端点x和y所对应的相邻结点集合$N(x) = \{ z \mid xz \in E \}$和$N(y)$)

证明 设$xy \in E(G)$. 设$N(x) = \{ z \mid z \in V(G) \wedge xz \in E(G) \}$. 则$y \in N(x) \wedge x \in N(y)$, 且$(N(x) - \{ x \}) \cap (N(y) - \{ x \}) = \varnothing$. 否则设$z \in N(x) - \{ x \} \cap N(y) - \{ x \}$, 则$xyzx$成一三角形，这与围长大于3矛盾. 又$\deg(x) = \deg(y) = k$，因此$|N(x)| = N(y) = k$, 这样$V \supseteq \{ x \} \cup \{ y \} \cup N(x) \cup N(y) = \{ x \} \cup \{ y \} \cup (N(x) - \{ x \}) \cup (N(y) - \{ x \})$. 因此$v(G) \geqslant 1 + 1 + k - 1 + k - 1 = 2k$.

§7 设G是非完全简单连通图，试证明对任意的结点$u \in V$，均存在$v, x \in V$使得$\{ u, x, v \}$导出的子图是P_3(3个结点的基本路径).

证明 $\forall u \in V$，需找到vx使得$uv \notin V$, 且$ux, xv \in E$. 考虑所有与u相邻的点$N(u)$. 若$N(u)$导出的子图不是完全图，即$N(u)$中不是每对结点相邻. 则存在$x, v \in N(v)$, 且$xv \notin E$，这样$\{u,\ x,\ v\}$导出的子图是P_3. 若$N(u)$导出的子图是完全图，即$N(u)$中每对结点均相邻, 则$N(u) \cup \{ u \}$导出的子图也是完全图. 而$N(u) \cup \{ u \} \subsetneq V$, 否则$G$是完全图与题设矛盾. 这样存在$y \in V - (N(u) \cup \{ u \})$. 且$y$与$N(u) \cup \{ u \}$有路通达. 这样存在$v \in V - (N(u) \cup \{ u \})$, 且$v$与$(N(u) \cup \{ u \})$中的某个点$x$相邻，而$v \notin N(u)$, 因此$ux, xv \in E \wedge uv \notin E$, 即$\{ u, x, v \}$ 导出的子图是P_3. □

§8 设G是简单图. 试证明若$u, v \in V$有两条不同的基本路径P和Q，则$P \cup Q$上一定存在一个基本回路.

证明 由于G是简单图，可简记u, v路为结点序列. 设 $P = x_0x_1 \cdots x_m\ (x_0 = u,\ x_m = v)$, $Q = uy_1y_2 \cdots y_n (y_0 = u, y_n = v)$. 由于$P$和$Q$不同，则存在$i \in \{0, 1, \ldots, \min\{m, n\}\}$, 使得$x_i = y_i \wedge x_{i+1} = y_{i+1} \wedge \forall j < i,\ x_j = y_j$. 考虑路$x_ix_{i+1} \cdots x_m$, 设$x_k$是$\{ x_{i+1}, \ldots, x_m \}$中下标最小的使得$x_k \in \{ y_i, y_{i+1}, \ldots, y_m \}$. 设$y_j = x_k$, 且$j \in \{ i, i+1, \ldots, m \}$. 则两路径$x_i \cdots x_k$和$y_i \cdots y_j$

的端点相同且内部没有相同的结点即$x_i \cdots x_k \cup y_i \cdots y_j$构成基本回路.

§9 用反证法证明若n个结点的图G有n条边，则G一定有圈.
证明 设G没有基本回路，则根据割边定理G的每个边都是割边. 这样删除一条边将增加一个连通分支，删除n个边后G将至少有$n+1$个连通分支. 与G的结点数n矛盾.

§10 设G是简单图，试用反证法证明G一定有条长度为δ的基本路径 (考虑最长路径小于δ).
证明 设$\delta = k$. 设最长路径$v_1 v_2 \cdots v_k$的长度为$k-1$. 而$\deg(v_k) \geqslant k$. 因此v_k 所引边的端点不可能全落在$\{ v_1, v_2, \ldots, v_{k-1} \}$, 即存在$v_{k+1} \notin \{ v_1, v_2, \ldots, v_{k-1} \}$ 且$v_k v_{k+1} \in E$. 这样$v_1 v_2 \cdots v_k v_{k+1}$是一条更长的路径. 矛盾.

§11 设u, v是n阶简单图G的两个结点，设$uv \in E$. 试证明G中含有边uv的三角形至少有$\deg(u) + \deg(v) - n$个. (提示：考虑集合$N(u) - \{ v \}$和$N(v) - \{ u \}$, 利用集合容斥原理)
证明 设$N(u)$是所有与u相邻的结点集. 定义$N'(u) = N(u) - \{ v \}$, $N'(v) = N(v) - \{ u \}$. 则$|N'(u)| = |N(u)| - 1 = \deg(u) - 1$, 同样$|N'(v) = \deg(v) - 1$. 而以$uv$为其中一边的三角形的第3点一定在集合$N'(u) \cap N'(v)$之中. 集合$N'(u) \cap N'(v)$的基数即是以$uv$为边的三角形的个数. 而由容斥原理有$|N'(u) \cup N'(v)| = |N'(u)| + |N(v')| - |N'(u) \cap N'(v)|$. 即$|N'(u) \cap N'(v)| = |N'(u)| + |N'(v)| - |N'(u) \cup N'(v)| \geqslant \deg(u) - 1 + \deg(v) - 1 - (n-2) = \deg(u) + \deg(v) - n$.

§12 设G是n阶简单图且$\delta \geqslant (n-1)/2$. 则G的直径不大于2.
证明 设u和v是G的任意两不同的结点. 若$uv \in E$，则u, v可达. 若$uv \notin E$. 考虑$N(u)$和$N(v)$. 若$w \in N(u) \cap N(v) \neq \emptyset$, 则存在路径$uwv$使得$u, v$可达. 而$|N(u) \cap N(v)| = |N(u)| + |N(v)| - |N(u) \cup N(v)| \geqslant (n-1)/2 + (n-1)/2 - (n-2) = 1$，即$|N(u) \cap N(v)$非空.

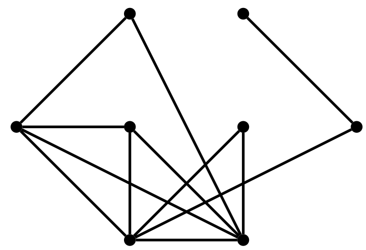
§13 设G是简单图且$\overline{G}$的直径大于或等于3，试证明G的小于或等于3.
证明 直接利用讲义例题即可. 设G的直径大于3，则$(\overline{G}) < 3$. 与题设矛盾.

§14 设有如下度序列，试判断它们是否可绘图，若可绘图，请画出相应的图.

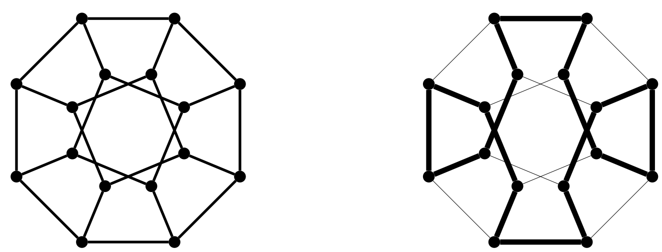
(1) $5, 5, 4, 3, 2, 2, 2, 1$;

(2) $5, 5, 5, 4, 2, 1, 1, 1$.

解 (1)是可绘图的, 其图如下所示；而(2)不是.



§15 试为下图找出一条Hamilton回路.



§16 设图G有Hamilton回路, 则对任意的结点子集合$S \subseteq V$有图$G-S$的连通分支数最多是$|S|+1$.

证明 设P是图G的一条Hamilton路. 在P上删除一个点，连通分支数最多加1. 这样$\omega(P-S) \leqslant |S|+1$. 而$P$是$G$的支撑子图，即$G$由$P$上加边得到, 而加边不可能增加连通分支数. 故$\omega(G-S) \leqslant \omega(P-S) \leqslant |S|+1$.

§17 试证明任意的竞赛图都存在Hamilton路径.

证明

(1) 设x_1x_2是任意有向边. 令$P=(x_1x_2)$.

(2) 若$P=n-1$. 则P是Hamilton路径，否则

(3) 设$P=x_1x_2\cdots x_m$. 则存在$y \notin V(P)$.

i. 若$y \longrightarrow x_1$, 则令$P=yx_1\cdots x_m$. 否则$x_1 \longrightarrow y$.

ii. 若存在$i \in \{1,2,\ldots,m-1\}$ 使得$y \longrightarrow x_{i+1}$，则令$x_1\cdots x_i y x_{i+1}\cdots x_m$. 否则$x_m \longrightarrow y$

iii. 令$P=x_1\cdots x_m y$.

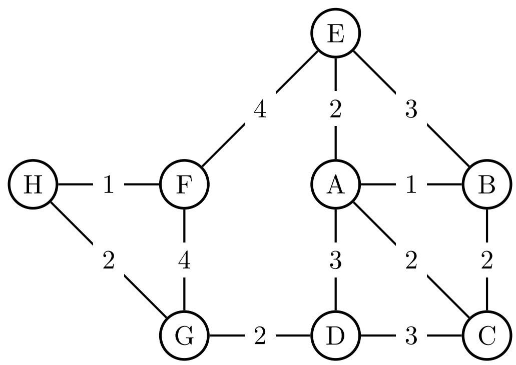
返回到(2).

按上述方法得到的路P包含了每个点，即Hamilton路.

§18 试证明$K_{n,n}(n \geqslant 2)$有$n!(n-1)!/2$条Hamilton回路.

证明 设X和Y是$K_{n,n}$的分区. 其均有n个结点. 固定$x \in X$作为Hamilton回路的始点, 则在Y上可选任意的一个点作为回路的后续点，有n个可选. 而在X上可选剩下的任意点为后续点，即有$n-1$个可选. 同样在Y上可选剩下的任意点作为回路的后续点, 即$n-1$个可选. 如此下去. 故有$n!(n-1)!$个回路. 由于正向与逆向经过x的回路是同一个回路，因此上数被多计数一倍，故总回路数为$n!(n-1)!/2$.

§19 用Dijkstra算法求下图结点H到其他结点的最短距离:

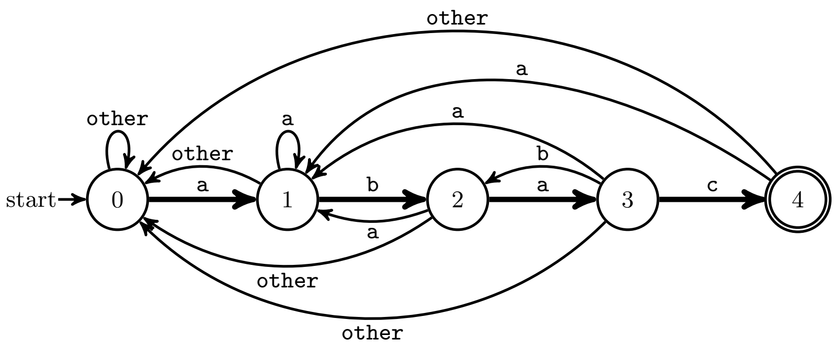


解

H	F	G	E	D	A	C	B	选择
0	∞	∞	∞	∞	∞	∞	∞	H
	$1(H)$	$2(H)$	∞	∞	∞	∞	∞	F
		$2(H)$	$5(F)$	∞	∞	∞	∞	G
			$5(F)$	$4(G)$	∞	∞	∞	D
			$5(F)$		$7(D)$	$7(D)$	∞	E
					$7(D)$	$7(D)$	$8(E)$	A
						$7(D)$	$8(E)$	C
							$8(E)$	B

$20 现需设计一个对任意的字符串查找是否有子串`abac`的函数`int match(char *s)`，如果发现有子串“abac”，则函数返回`1`，否则返回`0`. 常规算法是面对输入`acabac`, 首先用`acab`与需查找的串`abac`进行比较，若不成功再用`caba`, `abac`, 如此下去直到找到为止. 这需要对输入串进行多次扫描, 因而算法的效率不高. 不能用于对扫描时间要求特别苛刻的程序,如病毒的扫描等. 为此特设计如下状态图, 利用该图对输入仅一次扫描即可完成查找(详见KMP算法: `http://en.wikipedia.org/wiki/`

Knuth-Morris-Pratt_algorithm):



(1) 将上述状态图转换为流程图;

(2) 利用C语言的控制流结构实现函数`int match(char *s)`.

```
int match (char *s)
{
  char *cp = s;
  while ( *cp != 0 ) {
      /* state 0 */
    while (*cp++ == 'a')  { /* state 1 */
      for ( ; ; ) {
        if (*cp == 'b') cp++; /* state 2 */
        else break;
        if (*cp == 'a') { /* state 3 */
          cp++;
          if (*cp == 'c')  /* state 4 */
            return 1;
        }
      }
    }
  }
  return 0;
}
```