离散数学中英文对照及知识

chap 9:

9.1-9.3

the relation on the set A——from A to A:集合A上的关系（A，A）
connection matrice:连接矩阵
reflexive:自反性
irreflexive:非自反性
symmetric:对称性
antisymmetric:反对称性
transitive:传递性
the composite of R and S:矩阵的复合（S o R）——用矩阵乘法计算

定理

The relation R on a set A is transitive if and only if $R^n \subseteq R$,for n=1,2,3......
Inverse relation:关系的逆——定义法，求矩阵的转置
(R∪S)^-1 = $R^-1 \cup S^-1$

9.4

closure:闭包
I_A:对角线全是1(自反矩阵)

定理

Let R be a relation on A. There is a path of length n from a to b if and only if (a,b)∈ R^n

定义

The connectivity relation denoted by R^*,is the set of ordered pairs(a,b) such that there is a path (in R)from a to b:
$R^* = \sum_{n=1}^{\infty} R^n$

构造传递闭包：先写出连接矩阵，然后找(n,1)和(1,m),把(n,m)补上，再找(n,2)和(2,m)，把(n,m)补上,看到矩阵上就是先找第一列，再找第一行，然后在和左上角和右下角把构成方形的格子补上

9.5

Euler Path:欧拉通路，经过每条边
Euler Circuit:欧拉回路
Euler Graph:欧拉图——是欧拉回路的图
Hamilton path:哈密顿通路，经过每个点
Hamilton circuit(Hamilton cycle):哈密顿回路
Hamilton graph:哈密顿图
哈密顿图的实例：

〖Example 9〗 There are seven people denoted by A, B, C, D, E, F, G. Suppose that the following facts are known.

A--English (A can speak English.)

B--English, Chinese

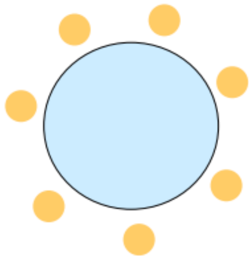
C--English, Italian, Russian

D--Japanese, Chinese

E--German, Italia

F--French, Japanese, Russian

G--French, German

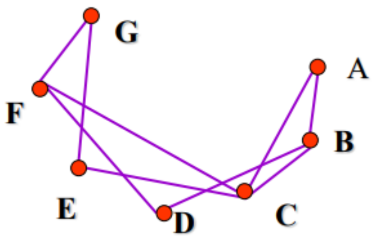


How to arrange seat for the round desk such that the seven people can talk each other?

Solution:

(1) Construct graph

V={A,B,C,D,E,F,G}, E={(u,v)|u,v can speak at least one common language.}



A--English (A can speak English.)

B--English, Chinese

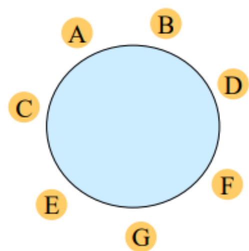
C--English, Italian, Russian

D--Japanese, Chinese

E--German, Italia

F--French, Japanese, Russian

G--French, German



(2) If there is a H circuit, then we can arrange seat for the round desk such that the seven people can talk each other.

H circuit: A,B,D,F,G,E,C,A

9.5

equivalence relations:等价关系：自反，对称，传递

【 Theorem 1】 Let R be an equivalence relation on a set A. The following statements are equivalent:

(1) aRb

(2) $[a]=[b]$

(3) $[a]\cap[b]\neq\phi$

Proof:

- **Show that (1) implies (2)**

$$[a]=[b]\Rightarrow([a]\subseteq[b])\wedge([b]\subseteq[a])$$

$$\left.\begin{array}{l} x\in[a]\Rightarrow(a,x)\in R \\ aRb\Rightarrow(a,b)\in R\Rightarrow(b,a)\in R \end{array}\right\}\Rightarrow(b,x)\in R\Rightarrow x\in[b]\Rightarrow[a]\subseteq[b]$$

(1) aRb

(2) $[a]=[b]$

(3) $[a]\cap[b]\neq\phi$

- **Show that (2) implies (3)**

$$\left.\begin{array}{l} [a]=[b] \\ \textbf{\textit{R} is reflexive} \Rightarrow \textbf{[\textit{a}] is nonempty} \end{array}\right\}\Rightarrow[a]\cap[b]\neq\phi$$

- **Show that (3) implies (1)**

$$\begin{aligned} [a]\cap[b]\neq\phi \quad &\Rightarrow\exists x\in[a]\cap[b] \\ &\Rightarrow(a,x)\in R,(b,x)\in R \\ &\Rightarrow(a,b)\in R \end{aligned}$$

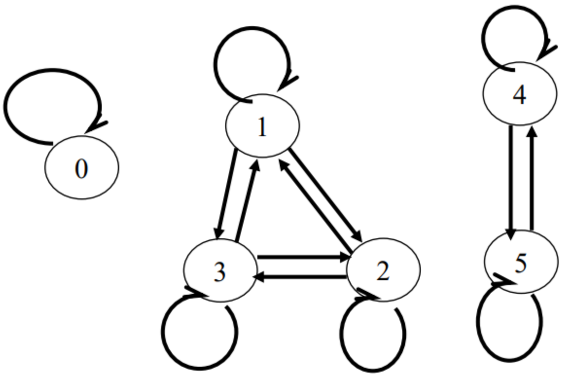
partition:分区

〖Example 4〗 Find the partition of the set A from R.

$A=\{0,1,2,3,4,5\}$,

$R=\{(0,0),(1,1),(2,2),(3,3),(1,2),(1,3),(2,1),(2,3),(3,1),(3,2),(4,4),(4,5),(5,4),(5,5)\}$

Solution:



$[0]=\{0\}$

$[1]=\{1,2,3\}$

$[4]=\{4,5\}$

$pr(A)=\{\,[0],\,[1],\,[4]\}$

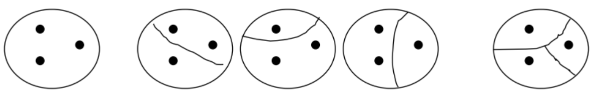
找A上的等价关系：相当于找A的划分

Question2:

|A|=3. How many different equivalence relations on the set A are there?

Solution:

an equivalence relation on a set A $\leftrightarrow$ a partition of A



9.6

partial orderings:偏序关系：自反，反对称，传递

poset:偏序集——表示为(S,R)，eg:a<=b (a,b) ∈ R:(S,R)是一个偏序集

comparable/incomparable:可比的，不可比的：元素都可以比大小

在不同的偏序集中，会使用不同的符号表示偏序，如$\leqslant$、$\subseteq$和$|$。然而，我们需要一个符号用来表示任意一个偏序集中的序关系。通常，在一个偏序集中，记号$a\preccurlyeq b$表示$(a,b)\in R$。使用这个记号是由于“小于或等于”关系是偏序关系的范例，而且符号$\preccurlyeq$和$\leqslant$很相似。（注意符号$\preccurlyeq$用来表示任意偏序集中的关系，并不仅仅是“小于或等于”关系。）记号$a\prec b$表示$a\preccurlyeq b$，但$a\neq b$。如果$a\prec b$，我们说“a小于b”或“b大于a”。

当a与b是偏序集$(S,\preccurlyeq)$的元素时，不一定有$a\preccurlyeq b$或$b\preccurlyeq a$。例如，在$(\mathcal{P}(\mathbf{Z}),\subseteq)$中，$\{1,2\}$与$\{1,3\}$没有关系，反之亦然，因为没有一个集合被另一个集合包含。类似地，在$(\mathbf{Z}^+,|)$中，2与3没有关系，3与2也没有关系，因为$2\nmid 3$且$3\nmid 2$。由此得到定义2。

定义 2 偏序集$(S,\preccurlyeq)$中的元素a和b称为可比的，如果$a\preccurlyeq b$或$b\preccurlyeq a$。当a和b是S中的元素并且既没有$a\preccurlyeq b$，也没有$b\preccurlyeq a$，则称a与b是不可比的。

例 5 在偏序集$(\mathbf{Z}^+,|)$中，整数3和9是可比的吗？5和7是可比的吗？

解 整数3和9是可比的，因为$3|9$。整数5和7是不可比的，因为$5\nmid 7$且$7\nmid 5$。◀

totally ordered/linearly ordered set:全序集：每对元素都是可比的 (也叫chain 链，形象理解成一条线)

total order/linear order:全序关系

◆ well-ordered

【Definition】 $(S, \preccurlyeq)$ is a **well-ordered set** if it is a poset suth that $\preccurlyeq$ *is a total ordering and* every nonempty subset of S has a least element.

For example,

$$(1) A=\{1,2,\ldots,n\},(A,\leq)$$

$$(2)(N,\leq)$$

$$(3) A=\{x \mid 0<x<1\},(A,\leq)$$

well-ordered set:良序集

lexicographic order:字典序

Hasse Diagrams：哈塞图

9.6.3 **哈塞图**

在有穷偏序集的有向图中，有许多边可以不必显示出来，因为它们是必须存在的。例如，考虑在集合$\{1,2,3,4\}$上的偏序$\{(a,b) \mid a\leqslant b\}$的有向图，见图 2a。因为这个关系是偏序的，所以它是自反的并且有向图在所有的顶点都有环。因此，我们不必显示这些环，因为它们是必须出现的。在图 2b 中没有显示这些环。由于偏序是传递的，所以我们不必显示那些由于传递性而必须出现的边。例如，在图 2c 中没有显示边(1，3)、(1，4)和(2，4)，因为它们是必须出现的。如果假设所有边的方向是向上的(如图 2所示)，我们不必显示边的方向，图 2c 没有显示边方向。

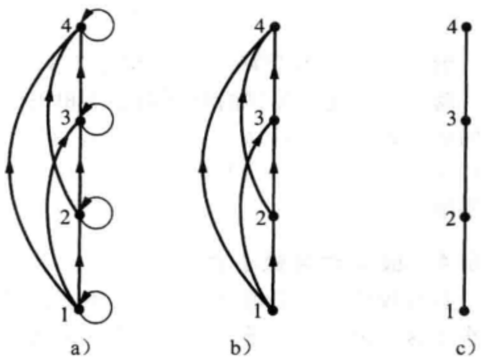


图 2　构造关于($\{1,2,3,4\}$，$\leqslant$)的哈塞图

一般说来，我们可以使用下面的过程表示一个有穷的偏序集。从这个关系的有向图开始。由于偏序是自反的，所以在每个顶点 a 都有一个环(a，a)。移走这些环。下一步，移走所有由于传递性必须出现的边。也就是说，对于元素 $z\in S$ 如果 $x\prec z$ 且 $z\prec y$，则移走所有这样的边(x，y)。最后，排列每条边使得它的起点在终点的下面(正如在纸上所画的)。移走有向边上所有的的箭头，因为所有的边"向上"指向它们的终点。

这些步骤是有明确定义的，并且对于一个有穷偏序集只有有限步需要执行。当所有的步骤执行以后，就得到一个包含足够的表示偏序信息的图，我们将在后面进行解释。这个图称为(S，$\preccurlyeq$)的**哈塞图**，它是用 20 世纪德国数学家赫尔姆·哈塞的名字命名的，哈塞广泛使用了这些图。

maximal and minimal element::极大元和极小元(多个或一个或没有)

greatest and least element:最大元和最小元(一个或没有，不是数字大小，看哈塞图)

upper and lower bounds:上界和下界(某一个子集的，多个或一个或没有)

least upper and greatest lower bounds(lub,glb):最小下界和最大上界(一个或没有)

lattice:格，该偏序集的每一对元素都有最大上界和最小下界

例 24　确定($\mathcal{P}(S)$，$\supseteq$)是否是格，其中 S 是集合。

解　设 A 和 B 是 S 的两个子集。A 和 B 的最小上界和最大下界分别是 $A\cup B$ 和 $A\cap B$，读者可自行证明。因此($\mathcal{P}(S)$，$\supseteq$)是格。

topological sorting:拓扑排序：每一步都选取剩余元素中的一个极小元，可以安排任务顺序

找A上的偏序关系：直接哈塞图画出来，一层一层的找

chap 10

10.1-10.5

无向边(undirected edge)：与集合$\{u, v\}$关联的边，其中u和v都是顶点。
有向边(directed edge)：与有序对(u, v)关联的边，其中u和v都是顶点。
多重边(multiple edges)：连接同样一对顶点的不同的边。
环(loop)：连接一个顶点与它自身的边。
无向图(undirected graph)：一组顶点以及连接这些顶点的一组无向边。
简单图(simple graph)：没有多重边和环的无向图。
多重图(multigraph)：可能包含多重边但不包含环的无向图。
伪图(pseudograph)：可能包含多重边和环的无向图。
有向图(directed graph)：一组顶点以及连接这些顶点的一组有向边。
有向多重图(directed multigraph)：可能包含多重有向边的有向图。
简单有向图(simple directed graph)：不含环和多重边的有向图。

相邻(adjacent)：若在两个顶点之间有边则它们是相邻的。
关联(incident)：若一个顶点是一条边的端点则那条边关联那个顶点。

图G中的匹配(matching in a graph G)：一组边的集合且任意两边都没有公共端点。
从V_1到V_2的完全匹配M(complete matching M from V_1 to V_2)：V_1中的每个顶点都是M中的边的端点的匹配。
最大匹配(maximum matching)：在图中所有匹配中包含边数最多的匹配。
孤立点(isolated vertex)：度为0的顶点。
悬挂点(pendant vertex)：度为1的顶点。
正则图(regular graph)：所有顶点都有相同的度的图。
图$G=(V, E)$的子图(subgraph of a graph $G=(V, E)$)：图(W, F)，其中W是V的子集而F是E的子集。
$G_1 \cup G_2$(G_1与G_2的并图)($G_1 \cup G_2$(union of G_1 and G_2))：图$(V_1 \cup V_2, E_1 \cup E_2)$，其中$G_1=(V_1, E_1)$和$G_2=(V_2, E_2)$。
邻接矩阵(adjacency matrix)：利用顶点的相邻关系来表示图的矩阵。
关联矩阵(incidence matrix)：利用边与顶点的关联关系来表示图的矩阵。
同构的简单图(isomorphic simple graphs)：对简单图$G_1=(V_1, E_1)$和简单图$G_2=(V_2, E_2)$来说，若存在从V_1到V_2的一一对应f，使得对所有属于V_1的v_1和v_2来说，$\{f(v_1), f(v_2)\} \in E_2$当且仅当$\{v_1, v_2\} \in E_1$，则$G_1$与$G_2$是同构的。
同构不变量(invariant for graph isomorphism)：同构的图都有或都没有的性质。
无向图里从u到v的通路(path from u to v in an undirected graph)：一条或多条边的序列$e_1, e_2, \cdots, e_n$，其中对$i=0, 1, \cdots, n$来说，e_i关联着$\{x_i, x_{i+1}\}$，其中$x_0=u$而$x_{n+1}=v$。
有向图中从u到v的通路(path from u to v in a graph with directed edges)：一条或多条边的序列$e_1, e_2, \cdots, e_n$，其中对$i=0, 1, \cdots, n$来说，e_i关联着(x_i, x_{i+1})，其中$x_0=u$而$x_{n+1}=v$。

connected graph:连通图
strongly connected directed graph:强连通图（强弱是只针对有向图的，去除所有方向的有向图称为有向图的基图，如果有向图是联通的，就是强连通图）
weakly connected directed graph:弱连通图（基图联通，有向图不连通）

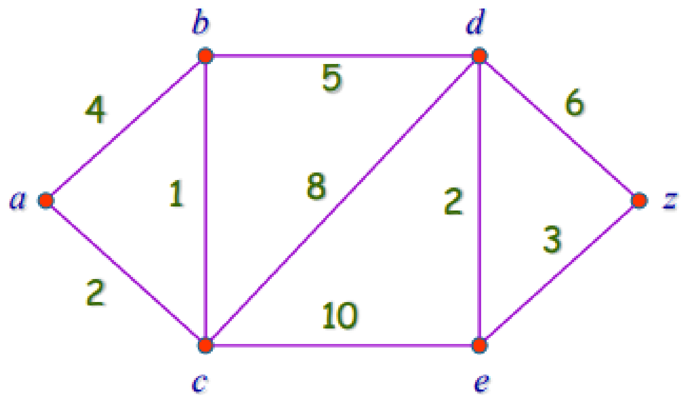
加权图(weighted graph):为各边指定数字的图。

最短通路问题(shortest-path problem):确定加权图中的通路以使得这条通路中的边的权之和在指定的顶点之间的所有通路上是最小值这样的问题。

旅行商问题(traveling salesperson problem):求访问图的每个顶点恰好一次的、总长度最短的回路的问题。

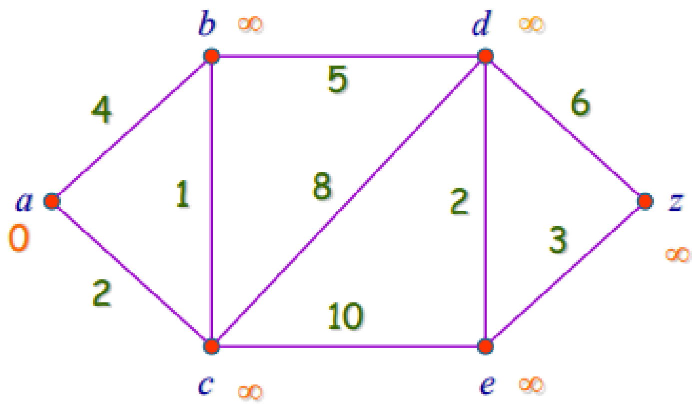
How Does Dijkstra's Algorithm Work?

【Example 2】 Find the length of the shortest path between *a* and *z* in the given weighted graph.



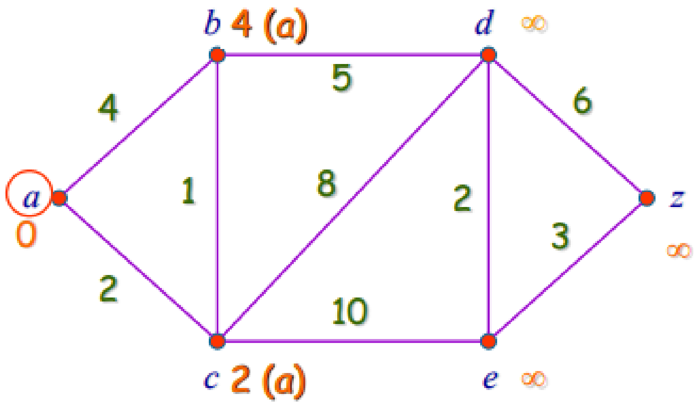
How Does Dijkstra's Algorithm Work?

Step 0



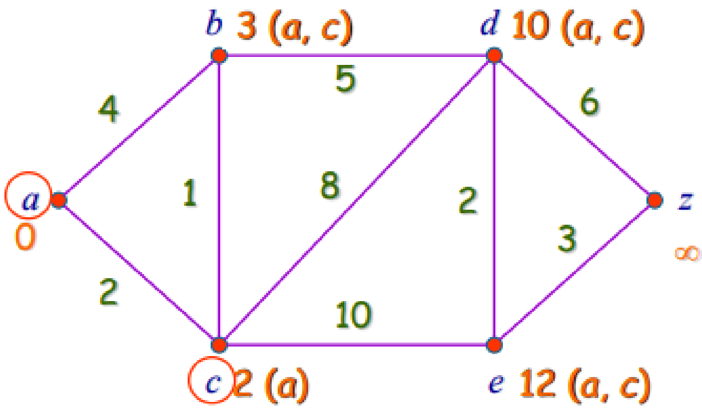
How Does Dijkstra's Algorithm Work?

Step 1



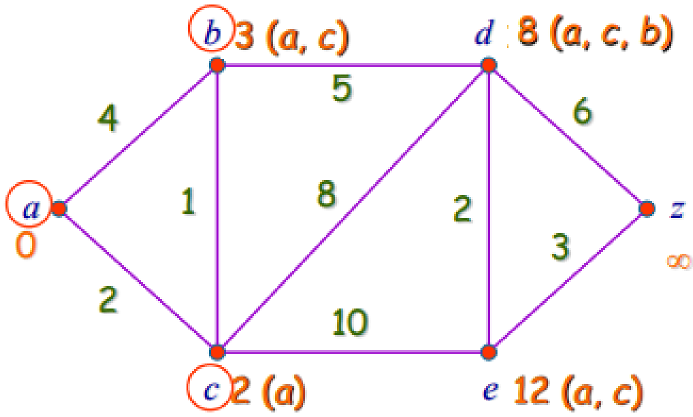
How Does Dijkstra's Algorithm Work?

Step 2



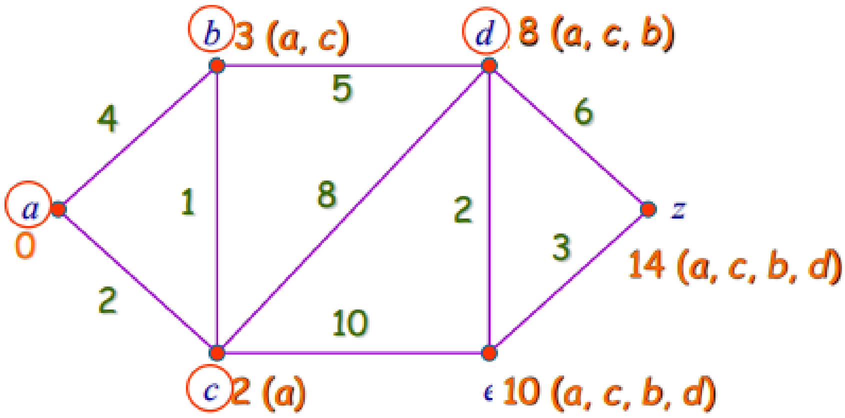
How Does Dijkstra's Algorithm Work?

Step 3



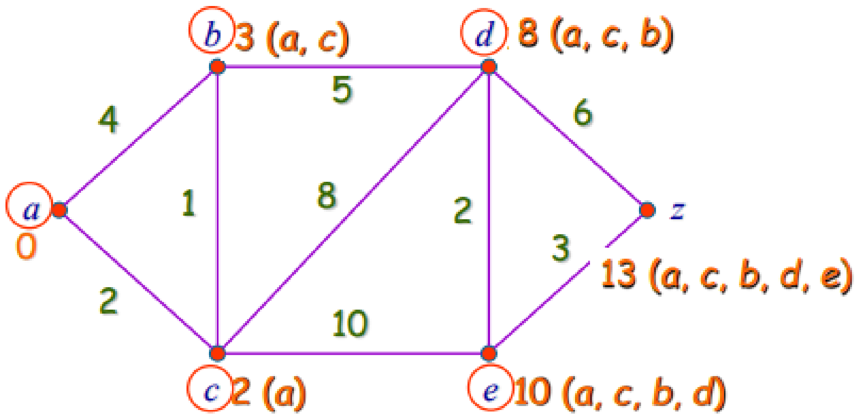
How Does Dijkstra's Algorithm Work?

Step 4



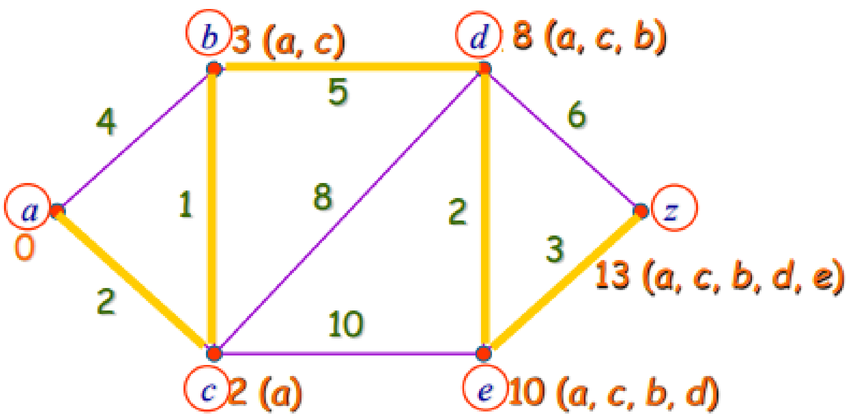
How Does Dijkstra's Algorithm Work?

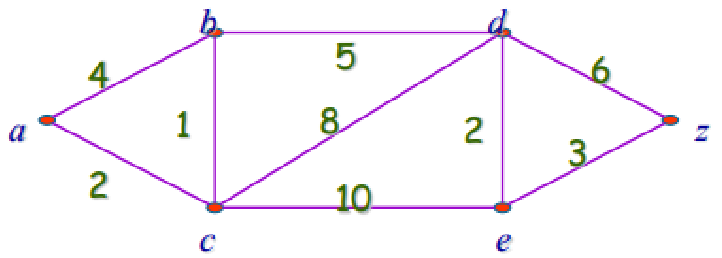
Step 5



How Does Dijkstra's Algorithm Work?

Step 6





Vertex	S	Link	L_0	L_1	L_2	L_3	L_4	L_5
a	1		0					
b	1	*a*→*c*	∞	4	3			
c	1	*a*	∞	2				
d	1	*a*→*c*→*b*	∞	∞	10	8		
e	1	*a*→*c*→*b*→*d*	∞	∞	12	12	10	
z	1	*a*→*c*→*b*→*d* → *e*	∞	∞	∞	∞	14	13

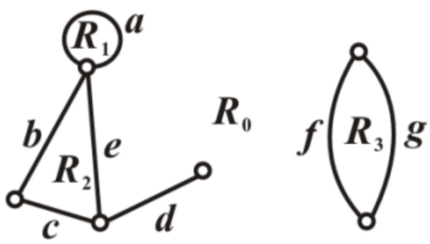
10.7

complete graph K_n:完全图，全部连满满
cycles C_n:圈
Wheels W_n:轮:在C_n中间加一个点，与其他点连线
n-Cubes Q_n:恰有一个bit不同的两个点连线，n指n个bit
边数edges的递推：$Q_n = 2 * Q_{n-1} + 2^{n-1}$
有向图：deg^-(v):terminate at v:终于v的边的个数
　　　deg^+(v):initiate at v:起于v的边的个数
planar graph:平面图
$K_{3,3}$和K_5不是平面图
(K_5指完全图，$K_{3,3}$指二分图)
region:区域
the boundary of region:区域的边界边(单向指出去的用两次)

〖Example 3〗 **There are 4 regions in the right graph.**

the boundary of region

R_1: *a*
R_2: *bce*
R_3: *fg*
R_0: *abcdde, fg*



deg(R_1) = 1
deg(R_2) = 3
deg(R_3) = 2
deg(R_0) = 8

10.8

初等细分(elementary subdivision)：删除无向图的边$\{u, v\}$而且添加新顶点w以及边$\{u, w\}$和边$\{w, v\}$。

同胚(homeomorphic)：若两个无向图是从同一个无向图通过一系列初等细分来获得的，则它们同胚。

图着色(graph coloring)：给图的顶点指定颜色，使得相邻的两个顶点没有相同的颜色。

着色数(chromatic number)：在图的着色中所需要的最少颜色数。

chap 10总结

结论

握手定理(The handshaking theorem)：设$G=(V, E)$是有m条边的无向图，则$2m=\sum_{v\in V}\deg(v)$

霍尔婚姻定理(Hall's marriage theorem)：带有二部划分(V_1, V_2)的二分图$G=(V, E)$中有一个从V_1到V_2的完全匹配当且仅当对于V_1的所有子集A，有$|N(A)|\geq|A|$。

在连通多重图中存在欧拉回路当且仅当每个顶点的度数都为偶数。

在连通多重图中存在欧拉通路当且仅当至多有两个度数都为奇数的顶点。

迪克斯特拉算法(Dijkstra's algorithm)：在加权图中求出两个顶点之间最短通路的过程(见10.6节)。

欧拉公式(Euler's formula)：$r=e-v+2$，其中r、e和v分别是平面图的平面表示的面数、边数和顶点数。

库拉图斯基定理(Kuratowski's theorem)：图是非平面图当且仅当它包含同胚于$K_{3,3}$或K_5的子图。(其证明超出本书范围。)

四色定理(The four color theorem)：每个平面图都可以用不超过4种颜色来着色。(其证明远远超出本书范围!)

欧拉公式：r=e-n(v)+2(r:regions:区域数，面数，e:edges:边数，n(v):vertice:顶点数)

可以数学归纳法证明

chap 11

树(tree)：没有简单回路的连通无向图。

森林(forest)：没有简单回路的无向图。

有根树(rooted tree)：具有一个规定的顶点(称为根)，使得从这个根到任意其他顶点有唯一通路的有向图。

子树(subtree)：树的子图，该子图本身也是一棵树。

有根树中顶点v的父母(parent of v in a rooted tree)：使得(u, v)是有根树的一条边的顶点u。

有根树中顶点v的孩子(child of a vertex v in a rooted tree)：以v作为父母的任何顶点。

有根树中顶点v的兄弟(sibling of a vertex v in a rooted tree)：与v具有相同父母的顶点。

有根树中顶点v的祖先(ancestor of a vertex v in a rooted tree)：在从根到v的通路上的任何顶点。

有根树中顶点v的后代(descendant of a vertex v in a rooted tree)：以v作为祖先的任何顶点。

内点(internal vertex)：具有孩子的顶点。

树叶(leaf)：没有孩子的顶点。

顶点的层(level of a vertex)：从根到这个顶点的通路的长度。

树的高度(height of a tree)：树里顶点的最大层数。

m 叉树(*m*-ary tree)：每个内点都有不超过 m 个孩子的树。

满 m 叉树(full *m*-ary tree)：每个内点都有恰好 m 个孩子的树。

二叉树(binary tree)：满足 $m=2$ 的 m 叉树(可以指定每个孩子作为父母的左子或右子)。

有序树(ordered tree)：对每个内点的孩子都线性地排序的树。

平衡树(balanced tree)：每个顶点都是在 h 层或 $h-1$ 层上的树，其中 h 是这棵树的高度。

二叉搜索树(binary search tree)：二叉树，在其中以元素对顶点进行标记，使得一个顶点的标记大于这个顶点的左子树里所有顶点的标记，并且小于这个顶点的右子树里所有顶点的标记。

决策树(decision tree)：性质如下的有根树，在其中每个顶点表示一次决策的可能输出，而树叶表示可能的解。

博弈树(game tree)：顶点表示博弈过程中的局面，边表示这些局面间的合法移动的有根树。

前缀码(prefix code)：一种编码，其中一个字符的编码永远不是另外一个字符的编码的前缀。

最小最大策略(minmax strategy)：第一个选手和第二个选手分别移动到具有最大值和最小值的孩子顶点所表示的局面的策略。

博弈树里顶点的值(value of a vertex in a game tree)：对于树叶来说，就是当游戏在这个树叶所表示的局面里结束时第一个选手的得分。对于分别在偶数或奇数层上的内点来说，就是这个内点的孩子的最大值或最小值。

树的遍历(tree traversal)：树的顶点的列表。

前序遍历(preorder traversal)：通过递归地定义的有序根树的顶点列表——列出根，接着列出第一棵子树，接着以从左到右的出现顺序列出其余子树。

中序遍历(inorder traversal)：通过递归地定义的有序根树的顶点列表——列出第一棵子树，接着列出根，接着以从左到右的出现顺序列出其余子树。

后序遍历(postorder traversal)：通过递归地定义的有序根树的顶点列表——以从左到右的出现顺序列出各子树，接着列出根。

中缀记法(infix notation)：从表示表达式(包括全套括号)的二叉树的中序遍历所获得的表达式形式。

前缀记法，或波兰记法(prefix(or Polish) notation)：从表示表达式的二叉树的前序遍历所获得的表达式形式。

后缀记法，或逆波兰记法(postfix(or reverse Polish) notation)：从表示表达式的二叉树的后序遍历所获得的表达式形式。

生成树(spanning tree)：包含图的所有顶点的树。

最小生成树(minimum spanning tree)：边的权之和最小的生成树。