《集合与图论》总复习

范围：集合论（1-4章）、组合数学（5-7章）、图论（8-11章）

考试形式：

1. 是非判断题：判断命题是否正确, 并说明理由；
2. 证明、计算题。

命题要求：基本知识点考核；综合知识考核；给出新概念，证明性质或者计算。

1. 非空集合A上不存在二元关系R，使得R既是A上的等价关系，又是A上的偏序关系。

（假）

反例：恒等关系。

2. 设（A，≤）是偏序集，∅≠B⊆A，若B有上界，则B必有上确界。

（假）

反例：（{2，3，24，36}，/）。

3. 设R是集合A上的二元关系

1）求A上包含R的最小等价关系E的表达式;

2）证明E的最小性;

3）以A={1, 2, 3, 4, 5, 6}, R={(1, 2), (1, 3), (4, 4), (4, 5)}为例验证你的结果.

（建议评分：15分，每小题5分）

/\* 解题分析：

求A上包含R的最小等价关系，就是求R的自反、对称和传递闭包。

因为st(R)ts(R)，所以E的表达式应该是E=tsr(R)=rts(R)，而E=str(R)=rst(R)是不成立的。

最小性结合闭包的定义进行证明。\*/

解：

1. E=tsr(R)=rts(R)

证明：

2）假设P是集合A上包含R的任一等价关系。

因为P是自反的，所以r(R)⊆P；

因为P是对称的，所以sr(R) ⊆P；

因为P是传递的，所以tsr(R) ⊆P；

所以E⊆P，从而保证了E的最小性。

3) E=tsr(R)=rts(R)=rt({(1, 2), (2, 1), (1, 3), (3, 1), (4, 4), (4, 5), (5, 4)})=r({(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2), (4, 5), (5, 4), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5)})= {(1, 2), (2, 1), (1, 3), (3, 1), (2, 3), (3, 2), (4, 5), (5, 4), (1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6)}

/\*考核知识点：等价关系\*/

4.计算中心安排领导和6名青年教师x1，x2，x3，x4，x5，x6值夜班，从周一到周六每个青年教师值一晚，周日领导值班。其中x1不安排在周一值班，x2不安排在周二值班，x3不安排在周三值班。问：共有多少种不同的安排值班方法？

解：全集为U，A 表示x1安排在周一值班的值班安排集合，B表示x2安排在周二值班的值班安排集合， C表示x3安排在周三值班的值班安排集合。

则安排值班方法=|U|-|A∪B∪C|=|U|-(|A|+|B|+|C|-|A∩B|-|A∩C|-|B∩C|+|A∩B∩C|)

|U|=6!=720

|A|=|B|=|C|=5!=120

|A∩B|=|A∩C|=|B∩C|=4!=24

|A∩B∩C|=3!=6

安排值班方法=426

/\*考核知识点：容斥原理，排列\*/

5．平面上有n(n≥2)个圆，任何两个圆都相交但无3圆共点，问这n个圆把平面划分成多少个不连通的区域？

解：n个圆把平面划分成an个不连通的区域

a0=1, a1=2, a2=4, an= an-1+2(n-1) (n≥2)

a1=n2-n+2 (n≥2)

/\*考核知识点：递推关系\*/

6. 矩形被分为3×7=21个正方形，每个正方形用红色或黑色着色。证明存在非简单子矩形（非1×*k*或*k*×1），4个角的正方形颜色相同。

解：设共有3行，7列牌。将同一列的两个同色的牌称为“同色对”。根据鸽笼原理，每一列至少有一个同色对，所以整个矩形含有7个同色对，每列一个。再根据鸽笼原理，至少有4个同色对的颜色完全相同，不妨设有4个红色的同色对。同色对在一列中有3种可能，再次根据鸽笼原理，这4个同色对至少有两个在列中具有相同的位置。这两个同色对的4个牌确定的矩阵满足条件。

/\*考核知识点：鸽笼原理 \*/

7. 用下述算法为简单图着色：

1. 以度数递减的顺序给出顶点的列表*v*1, *v*2, …, *vn*，使得d(*v*1)≥d(*v*2)≥ …d(*vn*)；
2. 把颜色1着色给顶点*v*1和在列表中不与顶点*v*1相邻的下一个顶点（若存在一个这样的顶点），并且继续给列表中每一个不与着颜色1 的顶点相邻的顶点着颜色1；然后，把颜色2 着色给列表中还没有着色的第一个顶点，并继续按上述步骤对列表中的顶点着颜色2；然后，以此类推，直到所有的顶点都被着色。

举例说明这一算法不是最优的，也就是说，这个算法所产生的着色所需的颜色数可能比某个图的色数大。

8. Prim算法

在Prim算法执行过程中，集合*A*中的边形成一棵树。初始时*A*为空；接下来每次添加到*A*的边都是使得树的权尽可能小的边。这个过程一直进行到生成树产生为止。图12.2-1给出了使用Prim算法计算最小生成树的过程，图中的阴影节点为出发点，即最小生成树的根。显然出发点不同，最小生成树的形态就不同，但边权和的最小值是唯一的。

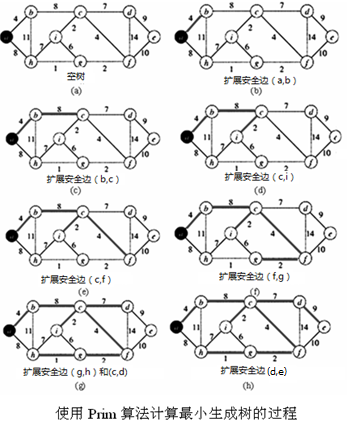


图12.2-1

下面，给出Prim算法的算法思想。

设*r*为出发节点；节点*i*是生成树外的节点，且*d*[*i*]为节点*i*与生成树相连的最短边长，即*d*[*i*]=，简称节点*i*的距离值；

在Prim算法执行过程中，所有不在树中的节点按照*d*值递增的顺序组成一个优先队列*Q*；

*f*[*u*]为树中*u*节点的父母。在Prim算法执行过程中最小生成树的边集*A*隐含地满足 *A*={(*u*, *f*[*u*])│*u*∈*V*-{*r*}-*Q*}。

当算法结束时，最小优先队列*Q*是空的，而最小生成树的边集*A*满足 *A*={(*u*, *f*[*u*])│*u*∈*V*-{*r*}} ，*ans=*。计算过程如下：

for (each *v*∈*G*(*V*)) //初始时除出发节点的距离值为0外，其它节点的距离值为∞，最小优先队列*Q* 包含所有节点，即所有节点未在生成树

{ *d*[*v*]=∞；*f*[*u*]=nil；}；

*d*[*r*]=0；*Q*=*G*(*V*)；

while (*Q*!=∅)

{ 在*Q*中取出一个*d*值最小的节点*u*； **//**节点*u*进入最小生成树

If（*u*!=*r*）*ans=ans+w*[*u*, *f*[*u*]]； //若节点*u*非树根，则累计边权和

for (each *v*∈*u*相邻的节点集) //更新每个与*u*邻接、且不在树中的节点*v*的*d*值和父指针

if ((*v*∈*Q*)&&(*w*[*u*, *v*]<*d*[*v*]))

{ *f*[*v*]=*u*；*d*[*v*]= *w*[*u*, *v*]； }

}；

输出最小生成树的权和*ans*；

由上可见，while循环次，每次循环需要对优先队列*Q*操作，算法的效率取决于*Q*的数据结构。如果采用数组实现*Q*的话，则每次while循环需要花O(*V*2)时间对*Q*进行排序，因此Prim算法的运行时间为O(*V*3)；如果采用小根堆实现*Q*的话，则可以在初始化部分增加一个建堆的操作，花费时间为O(*V*)。每次while循环，从堆*Q*中取一个*d*值最小的节点需要O(ln *V*)时间；内循环for总共执行次（因为所有邻接表的长度和为2），每次对堆*Q*中*d*值的更新需要O(ln *V*)时间。因此Prim算法的整个运行时间为O（*V*\*ln *V*+*E*\*ln *V*）。由于<2，因此运行时间的上限为为O（*V*\*ln *V*+*V*2\*ln *V*）。可见，Prim算法的效率取决于节点数，因此一般适用于稠密图。

练习：

1. 简单图的定向就是指定它的各边的方向，使得所得的图是强连通图。证明：若一个图有割边，则它不是可定向的。
2. 证明：马在国际象棋3×4的棋盘上可以遍历。
3. 如果一个带有*e*条边和*n*个顶点的连通简单平面图不包含长度为4或更短的回路。证明：若*n*≥4，则*e*≤（5/3）*n*-（10/3）。