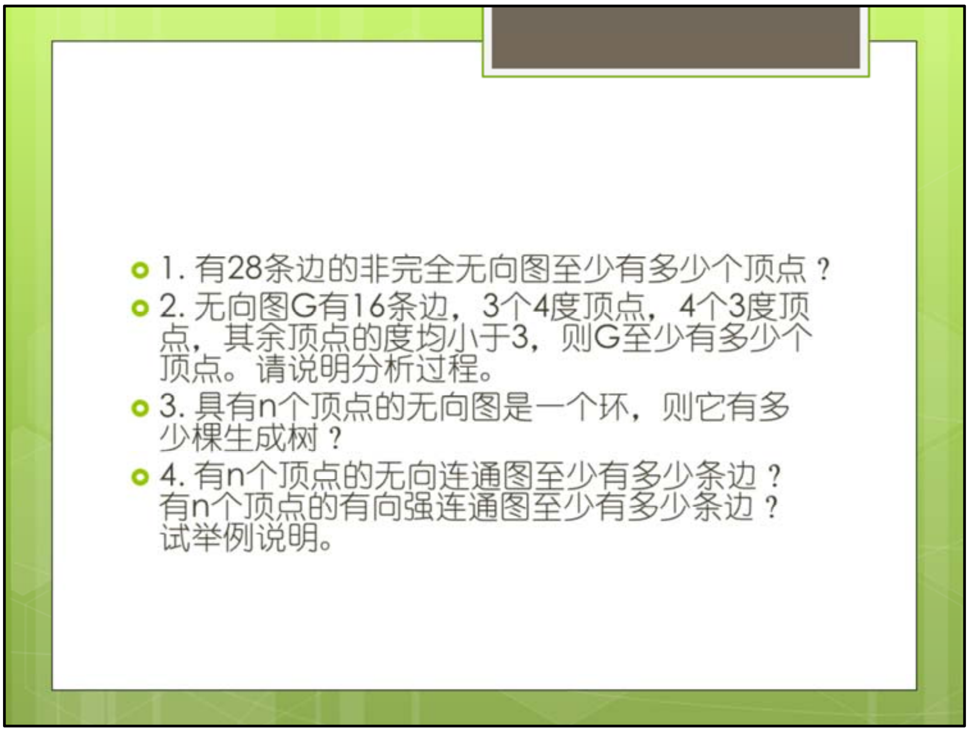
Chapter 8
练习题



答：

1. $n(n-1)/2 < 28$， $n > 8$， $n = 9$

2. $3 * 4 + 4 * 3 + 2 * n_2 + 1 * n_1 = 16 * 2$

=> $2n_2 + n_1 = 8$

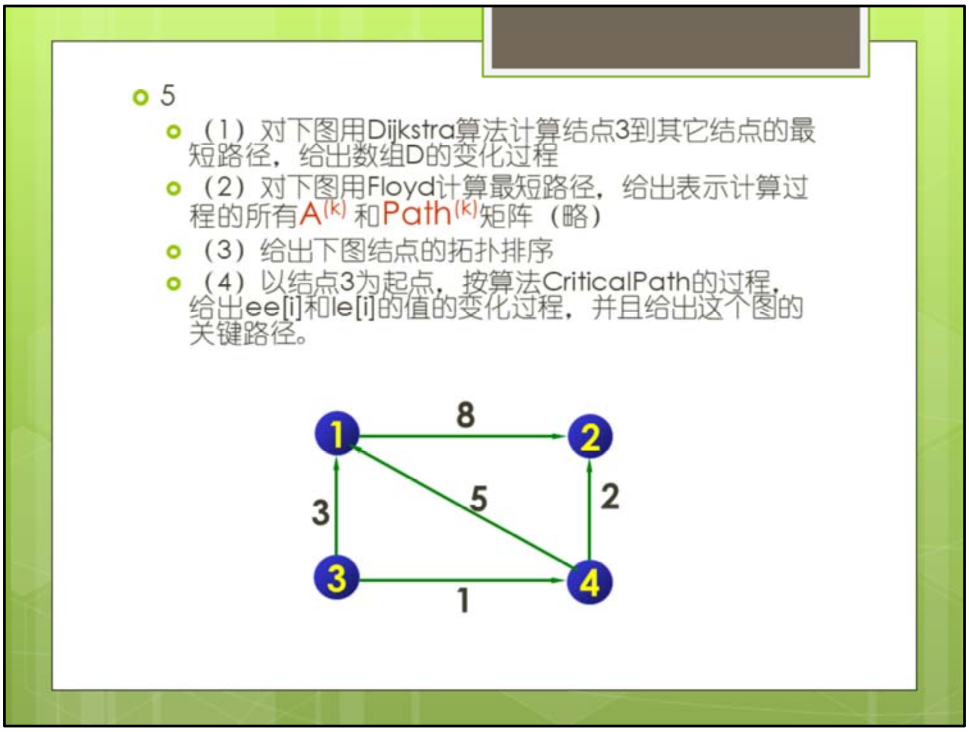
令 $n_2 = 4$, 得出$n_1 = 0$, 则总顶点数为 $4 + 3 + 4 = 11$。 通过手工画图可以把3个4度顶点和4个3度顶点一起构造成一个连通图，那么剩下的就是8个度，即4条边，形成一个4边形就可以了。

3.对每个顶点来说，去掉一条边就是一颗新的生成树。而且只能去掉一条边，因为树是边数最小的连通图。有的教材中把不指定树根的情况称为自由树，所以总共是n棵生成树

（自由树）。

但是如果把树根看作一个特殊的结点，则上述每棵生成树的树根可以有n种情况，则可以认为有n*n棵生成树。

4.有n个顶点的无向连通图至少有n－1条边（树）。有n个顶点的有向强连通图至少有n条边（环）。



（1）

D[1]=3， D[2]=Infinity, D[3]=0, D[4]=1

D[4]=1最小，用来改进其它路径：D[1]=3， D[2]=3, D[3]=0, D[4]=1

无法进一步改进，结束

（2）

A0= 0	8	*	*	A1=0	8	*	*	A2，A3不变	A4=0	8	*	*
*	0	*	*	*	0	*	*		*	0	*	*
3	*	0	1	3	11	0	1		3	3	0	1
5	2	*	0	5	2	*	0		5	2	*	0

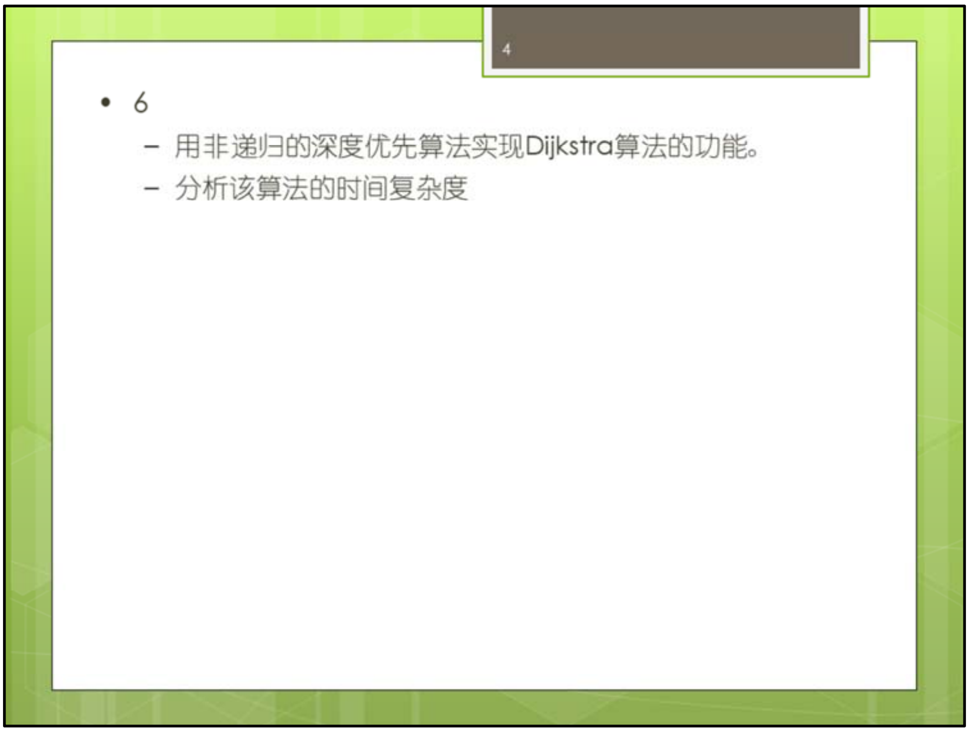
Path0=0	1	0	0	Path1=0	1	0	0	Path2，Path3不变	Path4=0	1	0	0
0	0	0	0	0	0	0	0		0	0	0	0
3	0	0	3	3	1	0	3		3	4	0	3
4	4	0	0	4	4	0	0		4	4	0	0

注意教材中的path数组不是常规的写法，因为其初始是全0，所以强调如果PPT与教材不一致时应该以PPT为准，不确定时问一下老师。

（3）

拓扑排序：3->4->1->2，

（4）
正推：E3=0, E4=1, E1=6， E2=14； 反推：L2=14, L1=6, L4=1, L3=0； 关键路径：3->4->1->2



就是非递归的遍历算法，与之前不同是不对已遍历过的结点设标志，而是把当前最短距离记录在遍历到的结点上，递归终止条件是不能改进遇到的结点的最短距离。

时间复杂度：
在普通的深度优先遍历过程中，每个顶点都要进一次栈且仅仅一次，并且会检查每一条边一次，在邻接矩阵表示时，检查所有的边的时间为$O(n^2)$,所以总时间为$O(n+n^2)=O(n^2)$。当图用邻接表储存时，那么当当顶点u进栈后，要寻找它的下一个邻接点v，时间为$o(e_1)$,其中e_1为u的邻接边个数, 因此总的时间复杂度为$0(n+e)=n+e_1+e_2+\ldots+e_n$；其中$e= e_1+e_2+\ldots+e_n$为无向图中边数。
由于这个练习中的算法仍然会访问已遍历过的结点，所以每次最坏情况会检查所有的边，所以时间复杂度=n+e+e+…+e=n+e*n，对于邻接矩阵访问所有的边需要n^2，所以对邻接矩阵时间复杂度$=n+e*n=n+n^2*n=O(n^3)$。