



- 1. 有28条边的非完全无向图至少有多少个顶点？
- 2. 无向图G有16条边，3个4度顶点，4个3度顶点，其余顶点的度均小于3，则G至少有多少个顶点。请说明分析过程。
- 3. 具有n个顶点的无向图是一个环，则它有多少棵生成树？
- 4. 有n个顶点的无向连通图至少有多少条边？有n个顶点的有向强连通图至少有多少条边？试举例说明。

答：

1.  $n(n-1)/2 < 28, \quad n > 8, \quad n = 9$

2.  $3 * 4 + 4 * 3 + 2 * n_2 + 1 * n_1 = 16 * 2$   
 $\Rightarrow \quad 2n_2 + n_1 = 8$

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

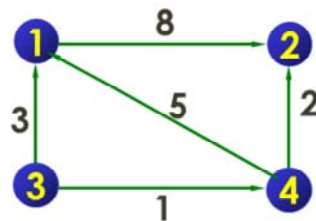
(自由树)。

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 $n \times n$ 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)。

5

- (1) 对下图用Dijkstra算法计算结点3到其它结点的最短路径，给出数组D的变化过程
- (2) 对下图用Floyd计算最短路径，给出表示计算过程的所有 $A^{(k)}$ 和 $Path^{(k)}$ 矩阵（略）
- (3) 给出下图结点的拓扑排序
- (4) 以结点3为起点，按算法CriticalPath的过程，给出 $ee[i]$ 和 $le[i]$ 的值的变化过程，并且给出这个图的关键路径。



(1)

$D[1]=3, D[2]=\text{Infinity}, D[3]=0, D[4]=1$

$D[4]=1$ 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)

$A_0 = \begin{matrix} 0 & 8 & * & * \\ * & 0 & * & * \\ 3 & * & 0 & 1 \\ 5 & 2 & * & 0 \end{matrix}$	$A_1 = \begin{matrix} 0 & 8 & * & * \\ * & 0 & * & * \\ 3 & \underline{11} & 0 & 1 \\ 5 & 2 & * & 0 \end{matrix}$	$A_2, A_3 \text{ 不变}$	$A_4 = \begin{matrix} 0 & 8 & * & * \\ * & 0 & * & * \\ 3 & \underline{3} & 0 & 1 \\ 5 & 2 & * & 0 \end{matrix}$
--	---	-----------------------	--

$Path_0 = \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 3 \\ 4 & 4 & 0 & 0 \end{matrix}$	$Path_1 = \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & \underline{1} & 0 & 3 \\ 4 & 4 & 0 & 0 \end{matrix}$	$Path_2, Path_3 \text{ 不变}$	$Path_4 = \begin{matrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 3 & \underline{4} & 0 & 3 \\ 4 & 4 & 0 & 0 \end{matrix}$
---	---	-----------------------------	---

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，不确定时间问一下老师。

(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

- 6
  - 用非递归的深度优先算法实现Dijkstra算法的功能。
  - 分析该算法的时间复杂度

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

$O(n+e)=n+e_1+e_2+\dots+e_n$ ；其中 $e=e_1+e_2+\dots+e_n$ 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 $=n+e+e+\dots+e=n+e*n$ ，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 $=n+e*n=n+n^2*n=O(n^3)$ 。