22.4-2

1. 先得到原图 G 的逆邻接链表图 G’，即得到每一节点的上一节点
2. 从 s 开始在原图上得到拓扑序列
3. 用 c（x）表示从 s 开始到 x 的简单路径数，c（s）=1，从 x 的上一层节点 v1，v2……到 x，c（x）=c（v1）+c（v2）+……，一直到节点 t，得到 s 到 t 的简单路径数 c（t）

22.4-5

1. 时间复杂度 O（V+E）原因
   a) 算法：首先统计每个点的入度，不断找每个入度为 0 的点删除并输出，与其有边的相邻节点入度-1，直到全部删除并输出。
   b) 这个算法只访问每个点和边一次，故复杂度 O（V+E）
2. 如果包含环路，环路中所有点都不会出现入度为 0 的点，需要先打破环路，才能输出拓扑序列

22.5-7

1. 算法：先运行强连通分量算法，用第二次 DFS 序给连通分量标号 1-n，生成 SCC 图，已知标号为 i 的 SCC 不会指向 i+1 的 SCC，那么只需要按顺序检查是否有（2,1）（3,2）……（n，n-1）的边。若都存在，则是半连通图；若有的边不存在，则不是。
2. 证明：若 SCC（i+1）到 SCC（i）有边，则 SCC（i+1）中的每个点到 SCC（i）的每个点都有路径；反之，则一定不存在路径

3. 时间复杂度：O(V+E)

22-3

a．(1) 欧拉回路=>入度=出度

对于所有点，若入度>出度，则必有一条边进入该点时无法找到出路

若入度<出度，则必有多余的出边未被访问

所以入度=出度

(2) 每个点的入度=出度=>有一条欧拉回路

从某一出发点 s 开始，若最后没有回到 s，必存在一条路径 s→v1→v2→……→vi，vi≠s，可以发现进入 vi 的次数大于出去的次数，所以 vi 必然不是结束点，同理直到回到 s，s 的进入次数=出去次数，这就是一条欧拉回路

b．(1) 从点 s 出发，顺序访问它未访问过的邻边

(2) 对于每一个新访问的点，同样顺序访问新的邻边

(3) 对于某个点没有新的邻边可以访问时，记录下最后一条边，回溯

(4) 所有边都访问过且回到起点 s 时，得到的路径就是欧拉回路