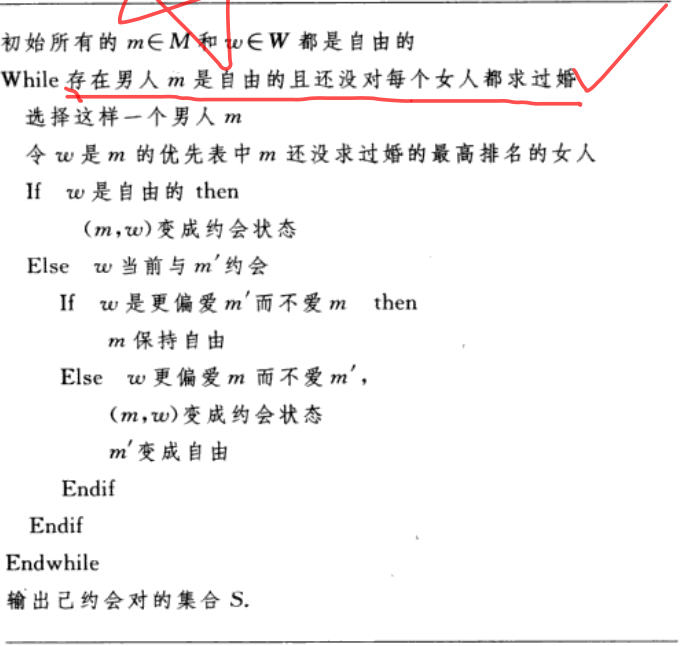
**第一章：**

**稳定匹配（GS算法）**：

对于一个示例可能有多个稳定匹配。

算法描述：一共分为两个部分，第一个部分是男生依次发出邀请，邀请的顺序是优先表上从高到低，然后女生接收邀请，女生接收邀请的逻辑是如果是自由，则直接接收男生的邀请，如果是匹配状态，则要根据女生自己的优先表选择优先级更高的男生。最后当所有人都在匹配状态算法结束。

伪代码：



算法分析；至多n方次停止。

算法返回是否是一个完美匹配？答：是，反证：如果不是完美匹配，说明有男生还是自由的而且他已经向所有女生求过婚了，说明这时候所有女生都在匹配状态，说明n个男生也都应该在匹配状态，这与该男生是自由的矛盾，即证。

算法返回是否是一个稳定的匹配？答：是，反证假设存在一个不稳定因素，（m，w）和（m’，w’），m更喜欢w’，w’也更喜欢m，现在根据这个匹配可知，m最后一个求婚的是w，但是由于m更喜欢w’，所以m一定在之前先向w’求过婚，但是被w’拒绝，可以知道w’拒绝的原因一定是有更喜欢的人和自己匹配，但是w’的匹配对象是m’，说明w’更喜欢m’，这里就产生了矛盾，命题即证。

**所有执行最后会得到相同的匹配，即男生拥有最佳有效伴侣的匹配，女生拥有最差有效伴侣。**有效伴侣：在一个稳定匹配中存在m和w的匹配，则w是m的一个有效伴侣。对于m来说，所有有效伴侣中在优先表中优先级最高的那个w就是最佳有效伴侣。

证明：反证：假设返回的匹配中有一个男生和不是他最佳有效伴侣的女生匹配，那么这个男生一定之前被他的最佳有效伴侣拒绝过，（**而且这是第一次一个男生被他的有效伴侣拒绝**）设这个男生是m，拒绝他的他的最佳有效伴侣是w，则说明w和另一个更喜欢的m’达成了匹配，现在考虑另一个稳定匹配中有（m，w），考虑这里的m’，假设是（m’，w’），由于刚才的拒绝是**第一次一个男生被他的有效伴侣拒绝，**所以m’还没有被他的有效伴侣拒绝过，这也说明（在第一个稳定匹配中），m’更喜欢w而不是w’，又因为w也更喜欢m’，这是稳定匹配中的一个不稳定因素，矛盾，命题即证。

**GS算法的时间复杂度是平方时间（优先表用数组存，状态用一维数组存，求婚用二维数组存，匹配对也用一维数组存。）**

扩展：稳定室友问题可能会没有稳定匹配解。

医院接受学生问题也可能没有解，不同点都有：两边数目不同、有不愿意匹配的对象组合、还可能有1v多的情况。

五大问题：

区间调度问题：贪心算法。

带权的区间调度问题：动态规划。

二分匹配问题：选择性回溯，网络流问题，找最大的边不想交的边集合。

独立集问题：NP完全问题

竞争的便利店选址问题：PSPACE问题，严格难于NP完全问题。

**带解答的练习**：

带禁止对的稳定匹配问题：只需修改算法的while循环条件，加上一个排除掉禁止对的条件。

**练习1、2**get

**第二章：**

效率：一个算法在真实的输入实例上运行的快，那么这个算法就是有效的。/如果一个算法与蛮力搜索相比，在最坏的情况下，可以达到质量上更好的性能，算法就是有效的。/如果一个算法有多项式运行时间，则这个算法是有效的。

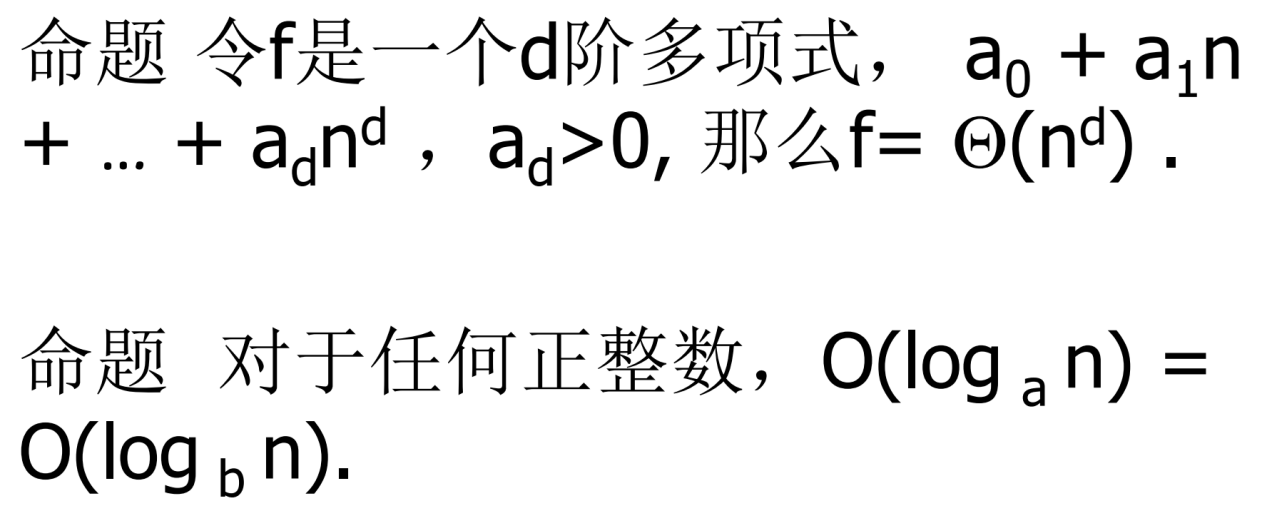
多项式有界。

渐近上界：O，不是一个准确的增长率只是一个界限。

渐近上界：Ω

渐近的紧的界：Θ，当渐近上界等于渐近下界的时候，那么这个界也是渐近紧的界。

具有传递性，可加性。

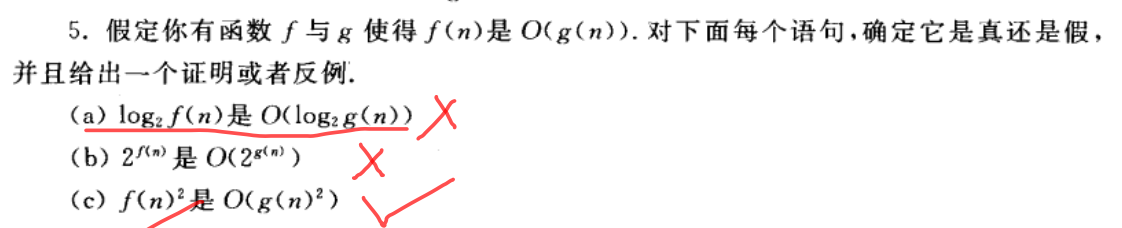


对数增长比多项式慢，多项式增长比指数慢。

压线性时间、线性时间、线性×对数时间（快排、FFT）、平方时间、立方时间、n的k次幂时间（给定规模k，寻找是否有独立集）、指数时间（最大独立集）、n阶乘时间（稳定匹配问题的蛮力搜索、二分匹配的蛮力搜索）

**带解答的练习**：get

**习题：3、5、6get、8（分治get）**

****

A）

B）

C）

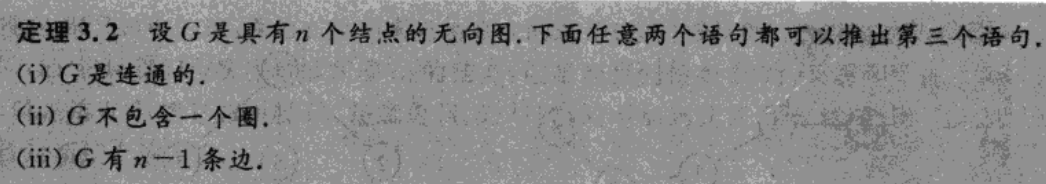
**第三章：**

一组点的序列就是一条路径，当所有节点都不相同的时候，这条路径是简单的。

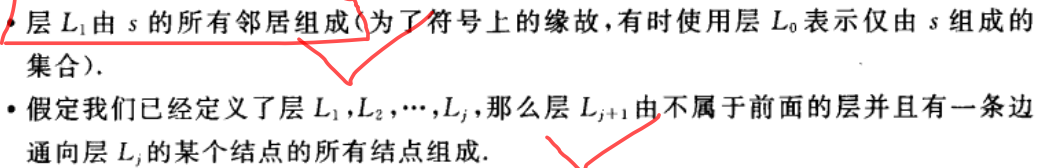
无向图的连通和有向图的弱连通（忽略边的方向后是连通的）、强连通（双向连通）

一个圈：一个路径节点序列仅仅头尾节点相同。

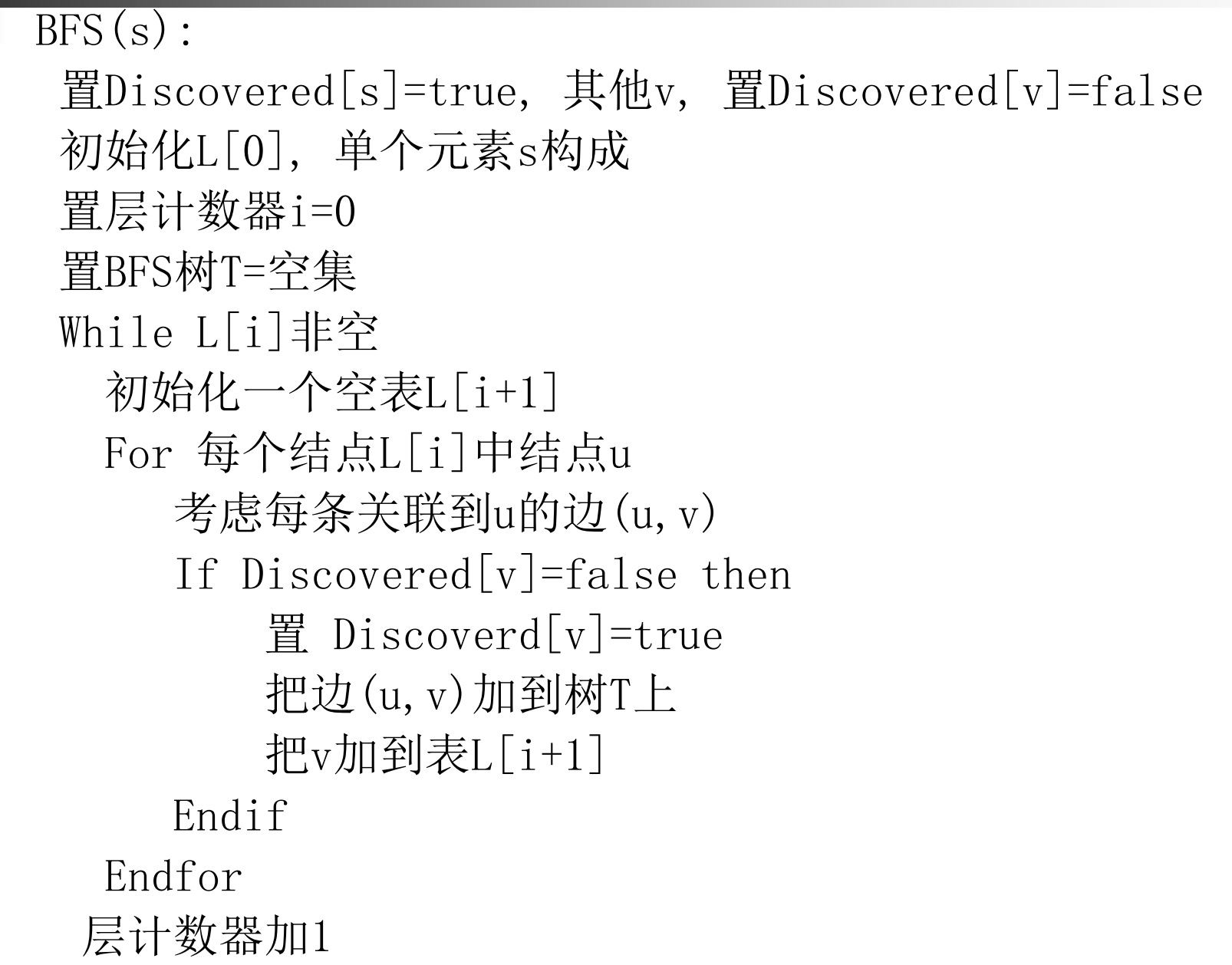
树：无圈的连通无向图，n个节点-（n-1）条边。



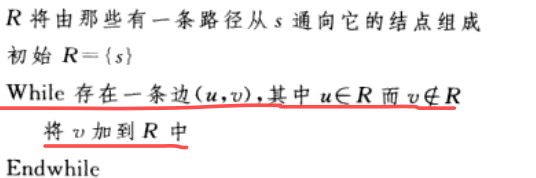
BFS：

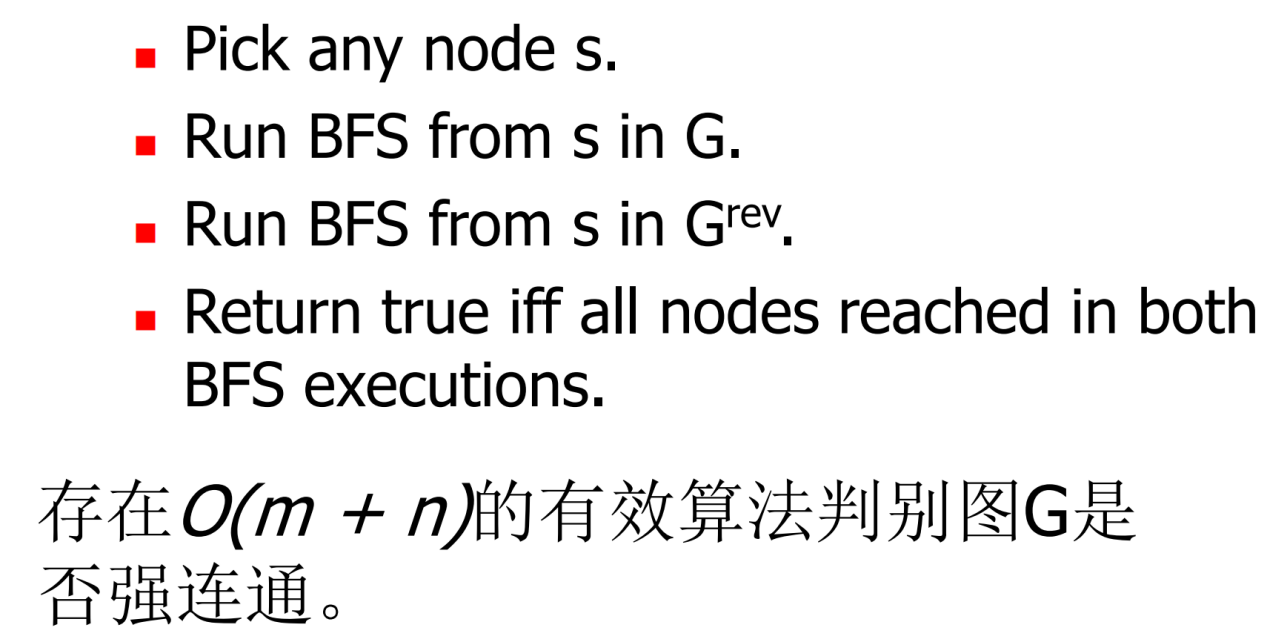


伪代码：



产生一个宽度有限搜索树，每层节点距离s的路径长度固定，事实上这些节点是图关于s的连通分支，那么下面有一个算法可以探索出关于s的连通分支：（无向图：两个节点所属的连通分支要么相等要么不交；有向图：～强连通分支～，G的rev是反向图）



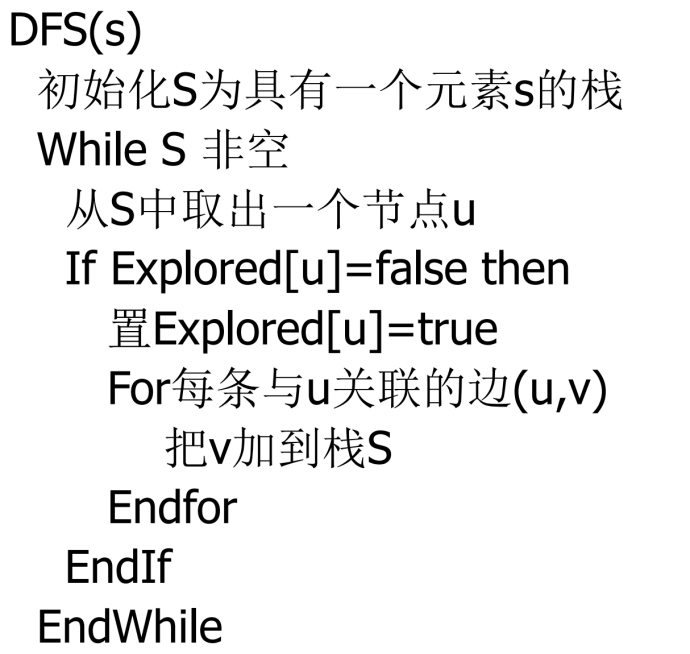


如果图是按邻接链表给出，BFS的时间复杂度是O（m+n），其中会访问所有边两次，另外对于节点，需要n的线性时间来管理是否访问过数组。

如果存在边（u，v）则u，v在树中相差层数至多是1.

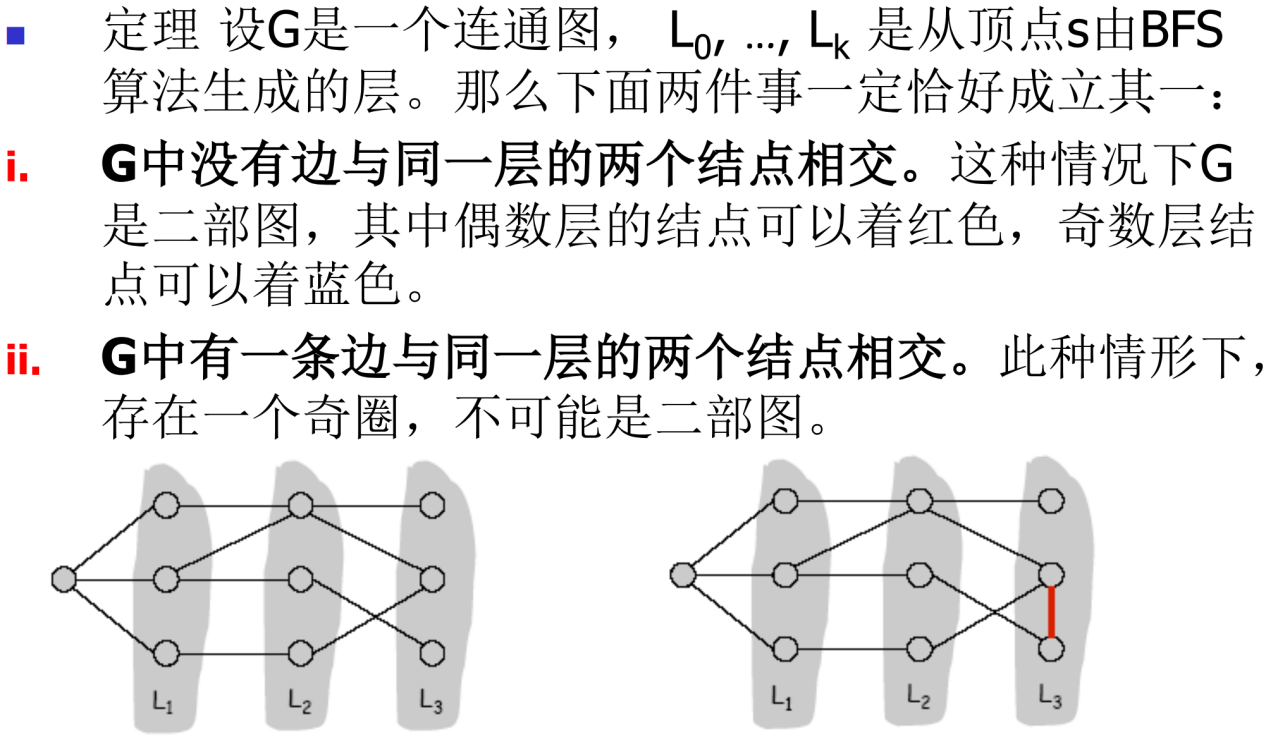
DFS：

伪代码：



二分性测试；

如果一个图是二部图，则这个图不包含奇圈。



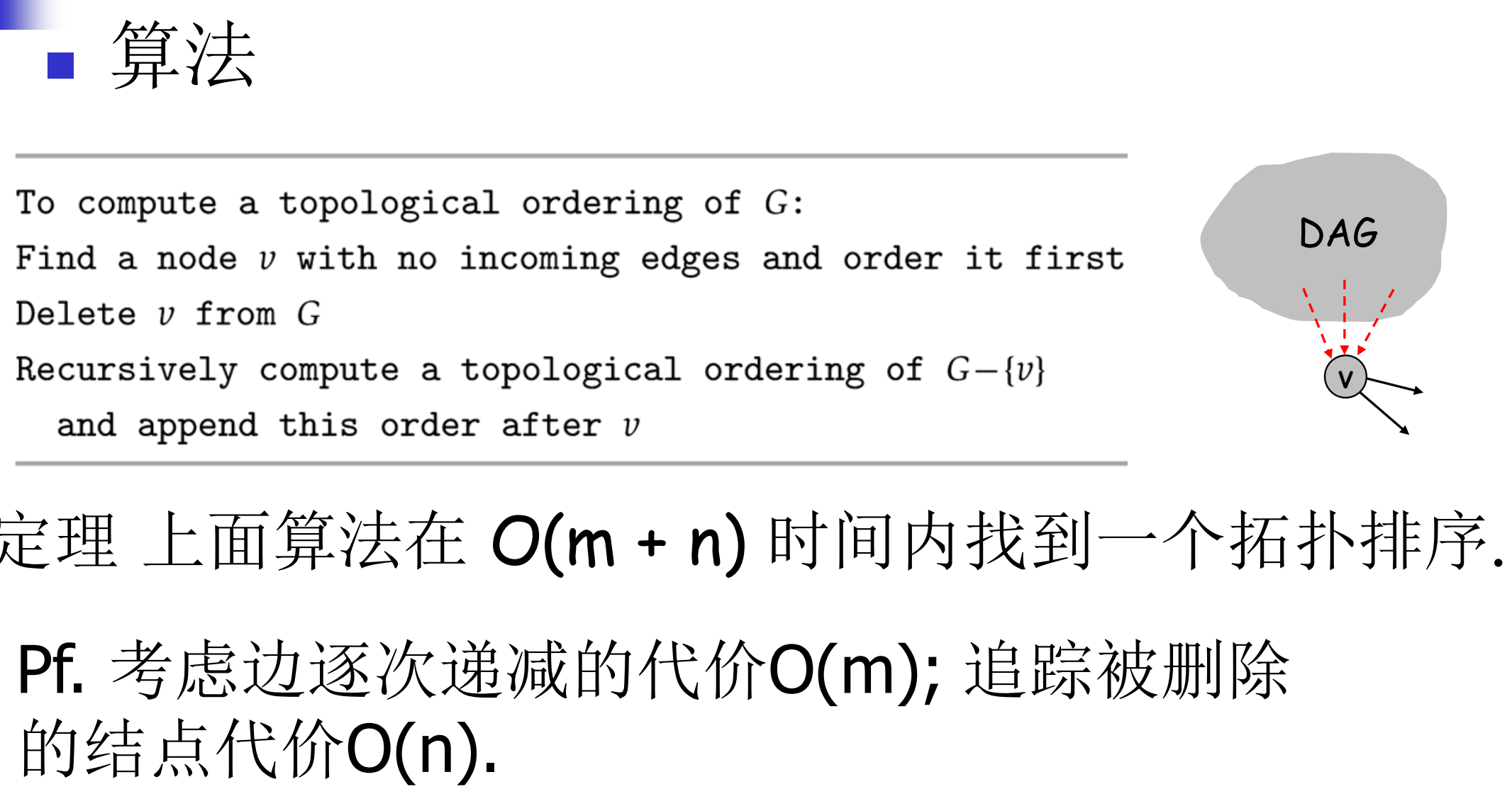
i情况的话，只需要红蓝交替着色即可。ii情况，说明存在奇圈。

DAG：

如果一个有向图有一个拓扑排序的**充要条件**是这个图为DAG。

算法设计分析；

引理：DAG中存在一个入度为0的节点。如果不存在一个入度为0的节点就说明该图不是一个DAG。



习题第一题get