Week 12

软73 沈冠霖 2017013569

May 20, 2019

1 T1

证明：如果有$L_1 \leq pL_2, L_2 \leq pL_3$，则必定可以找到函数$f(x), g(x)$，使得$\forall x, x \in L_1 \leftrightarrow f(x) \in L_2$，$\forall x, x \in L_2 \leftrightarrow g(x) \in L_3$。则任取一个$x_0 \in L_1$，必定可以找到$y_0 = f(x_0) \in L_2, z_0 = g_(y_0) \in L_3$，也就有$g(f(x_0)) = z_0$。同时，任取一个$z_1 \in L_3$,必定可以找到$y_1 \in L_2, g(y_1) = z_1, x_1 \in L_1, f(x_1) = y_1$，也就有$g(f(x_1)) = z_1$。这样，取函数h(x)=g(f(x)),就有$\forall x, x \in L_1 \leftrightarrow g(f(x))] \in L_3$。

2 T2

证明：这个问题应该转化为判定问题：给定图G和值k，是否存在一个G中简单回路总长度为k。

1.给定一个图X和一个代表回路顶点序列的串Y，只需要遍历Y，如果能构成长度为k的简单回路则成功，否则如果不是回路，或者不是简单回路，或者长度不是k，则失败。算法A(X,Y)=1复杂度为O(n)。

2.用哈密顿回路问题进行归约。定义函数$f(G) = (G, |V|)$（如果这个问题不带权，那么就这样了。否则把每条边权值设为1）。首先，如果G满足存在哈密顿回路，则这个图中必定存在一个简单回路-这个哈密顿回路，长度为$|V|$。反之，如果G不满足存在哈密顿回路，则一定不存在一个长度为$|V|$的简单回路。因此成功构造了满足要求的函数f。

3.f只需要把G和$|V|$复制过去，复杂度是O(n)，甚至O(1)，是一个多项式复杂度算法。

3 T3

a 定义：给定图G和非负整数k，是否存在一个大小为k的集合$V' \subset V$，使得V'满足独立性。

证明：1.可以找到一个判定算法A(X,Y),令X为给定的图G和非负整数k，Y'为这个图大小为k的子点集，只需要遍历G中所有边，如果每条边都最多与Y'最多一个点相关，则成立。否则不成立。遍历时间复杂度为$\theta(m) = O(n^2)$，每次遍历判断2次（常数次）,因此算法$A = O(n^2)$,是多项式算法，$L \in NP$。

2.用NPC问题–团问题进行归约。定义函数$f(G, k) = (\overline{G}, k)$。而如果图G和

值k满足团问题，也就是能找到一个子点集V'，使得V'中任意两个不同的点之间都有边，则对于同一个V'，在$\overline{G}$中，任意两个不同的点之间都没有边，也就是不存在$\overline{G}$中的一条边，其关联着V'中两个不同节点，也就是V'是$\overline{G}$大小为k的独立集。反之，如果存在一个子点集V'是$\overline{G}$大小为k的独立集，则在$\overline{G}$中，V'间任意两个不同的点都没有边，则G中V'间任意两个不同的点都有边，(G,k)满足团问题条件。因此找到了这样一个正确的函数f(G,k)，把团问题归约到独立集问题。 3.f(G,k)的操作是把邻接矩阵的每个值取非，时间复杂度是$\theta(n^2)$，因此是多项式复杂度算法。

b 因为子集V'大小最小为0，最大为$|V|$,因此从k=$|V|$开始，由大到小遍历$|V|$到0之间所有整数，如果找到第一个成功的k_0就停止。时间复杂度为$\theta(|V|)$。

c 1.因为每个点度数为2，因此一共有n条边。而且连通分支的个数和回路个数相等。
2.每个连通分支都是一整个回路。因为在每个连通分支中，每个点度数是2，因此每个连通分支的边数和点数一样，因此有且仅有一个回路。而因为每个点度数是2，如果这个连通分支不是一个回路的话，必定存在一个点，其度数为1，矛盾。
3.因此我的算法如下：遍历每个回路，也就是连通分支，如果回路长是1，就把这个点加入集合。如果回路长是偶数，也就是2n，就每隔一个节点把节点加入集合，一共加入n个节点。如果回路长是奇数，也就是2n+1，就每隔一个节点把节点加入集合，一共加入n个节点。根据回路的性质，这个算法正确，复杂度为$\theta(m) = \theta(n)$。

d 1.先求出最大匹配大小为$|M|$，则不在最大匹配对中的节点有$|V| - 2|M|$个，把这些加入独立集合S内。因为这些节点一定互相没有边，否则构成新的匹配，与最大匹配矛盾。
2.每个最大匹配对中必定可以找到一个节点加入S中。如果第i个最大匹配对中，两个节点a,b都不可以加入S，则必定有边(a,b),(a,x),(b,y),而且$x, y \in S$。那么$(a,x),(b,y) \notin M, (a,b) \in M$，构成了一条增广路径，与最大匹配没有增广路径矛盾。因此至少一个节点可以加入S。
3.这样找到的是最大匹配：因为这样找到的$|S| = |V| - |M|$，如果有比这个还大的集合，则根据鸽巢原理，必定有集合中两个点属于同一个匹配对，则这两个点间有边，矛盾。
4.因此我的算法如下：先按照26.3的算法求出最大匹配，然后把不在最大匹配里的点全部加入S，之后遍历每个最大匹配对，在每个匹配对里找到一个能加入集合S的点。求最大匹配的算法复杂度O(VE),匹配对有O(V)个，在每个匹配对里加入一个节点最多遍历E条边，因此总共复杂度为O(VE)。