算法设计与分析作业七

作者：吴润泽　　**学号：**181860109

Email：181860109@smail.nju.edu.cn

2020 年 6 月 4 日

目录

Chapter 15

problem 15.1

(1)

CLIQUE

优化问题

- 输入实例：无向图 G
- 优化问题：求图 G 中最大团的大小

判定问题

- 输入实例：无向图 G；参数 k
- 判定问题：图 G 中是否存在大小为 k 的团

KNAPSACK

优化问题

- 输入实例：n 个物品，其大小分别为 $s_1, s_2, \cdots, s_n$，每个物品的价值为 $c_1, c_2, \cdots, c_n$；参数 C
- 优化问题：背包中装若干物品，使得背包中物品的大小之和不超过 C 的物品价值和的最大值

判定问题

- 输入实例：n 个物品，其大小分别为 $s_1, s_2, \cdots, s_n$，每个物品的价值为 $c_1, c_2, \cdots, c_n$；参数 k 和 C
- 判定问题：是否可以在背包中装若干物品，使得背包中物品的大小之和不超过 C，且价值之和不低于 k

INDEPENDENT-SET

优化问题

- 输入实例：无向图 G
- 优化问题：求图 G 中最大独立集的大小

判定问题

- 输入实例：无向图 G；参数 k
- 判定问题：图 G 中是否存在大小为 k 的独立集

VERTEX-COVER

优化问题

- 输入实例：无向图 G
- 优化问题：求图 G 中最小点覆盖的大小

判定问题

- 输入实例：无向图 G；参数 k
- 判定问题：图 G 中是否存在大小为 k 的点覆盖

(2)

CLIQUE

优化问题多项式时间可解 设其最大团大小为 n，对于其判定问题，输入参数为 k。如果 $k \leq n$，则判定结果为 $true$，否则结果为 $false$。因此判定问题也是多项式时间可解。

判定问题多项式时间可解 假设图 G 点数为 n，则分别取判定问题输入参数 $k = 1, 2, \cdots, n$，当 $k < m$ 时结果均为 $true$，$k = m$ 时结果为 $false$，则图 G 的最大团大小即为 $m - 1$，则最多进行了 n 次的判定问题求解。由于多项式的计算封闭性，优化问题也为多项式时间可解。

KNAPSACK

优化问题多项式时间可解 设对于给定的若干物品，以及容量 C'，其最大的物品价值和为 m，对于其判定问题，令输入参数 $C = C'$，其另一输入参数 k。如果 $k \leq m$，则判定结果为 $true$，否则结果为 $false$。因此判定问题也是多项式时间可解。

判定问题多项式时间可解 对于给定容量 C，分别取判定问题输入参数 $k = sum(1), sum(2), \cdots, sum(n)$，$sum(i)$ 为前 i 个物品价值总和，当 $k < m$ 时结果均为 $true$，$k = m$ 时结果为 $false$，则背包中物品的大小之和不超过 C 的物品价值和的最大值即为 $m - 1$，则最多进行了 n 次的判定问题求解。由于多项式的计算封闭性，优化问题也为多项式时间可解。

INDEPENDENT-SET

优化问题多项式时间可解 设其最大独立集大小为 n，对于其判定问题，输入参数为 k。如果 $k \leq n$，则判定结果为 $true$，否则结果为 $false$。因此判定问题也是多项式时间可解。

判定问题多项式时间可解 假设图 G 点数为 n，则分别取判定问题输入参数 $k = 1, 2, \cdots, n$，当 $k < m$ 时结果均为 $true$，$k = m$ 时结果为 $false$，则图 G 的最大独立集大小即为 $m - 1$，则最多进行了 n 次的判定问题求解。由于多项式的计算封闭性，优化问题也为多项式时间可解。

VERTEX-COVER

优化问题多项式时间可解 设其最小点覆盖大小为 n，对于其判定问题，输入参数为 k。如果 $k \geq n$，则判定结果为 $true$，否则结果为 $false$。因此判定问题也是多项式时间可解。

判定问题多项式时间可解 假设图 G 点数为 n，则分别取判定问题输入参数 $k = n, n - 1, \cdots, 1$，当 $k \geq m$ 时结果均为 $true$，$k < m$ 时结果为 $false$，则图 G 的最小点覆盖大小即为 m，则最多进行了 n 次的判定问题求解。由于多项式的计算封闭性，优化问题也为多项式时间可解。

problem 15.2

证明 即证明如果有 $L_1 \leq_p L_2$，且 $L_2 \leq_p L_3$ 则有 $L_1 \leq_p L_3$。因为 $L_1 \leq_p L_2$，则问题 L_1 可以通过多项式时间转换函数 T_1 归约到问题 L_2，同理，问题 L_2 可以通过多项式时间转换函数 T_2 归约到问题 L_3。由多项式计算封闭性可知，L_1 可以通过多项式时间转换 T_3 归约到问题 L_3，因此 $L_1 \leq_p L_3$。

problem 15.4

设原始序列为 $a_1, a_2, \cdots, a_n$，选择输入参数 k，代表选择第 k 大。
排序归约到选择 对于排序的原始序列直接设为传入选择算法的序列即可，输入转换代价为常数，对序列进行 n 次选择，输入参数 $k = 1, 2, \cdots, n$，即依次选择第 i 大元素将其放在目标序列的对应位置。n 次选择后，即可得到已经排好序的序列，将其输出即可。

选择归约到排序 对于选择的原始序列直接设为传入排序算法的序列即可，输入转换代价为常数，然后根据选择算法的输入参数 k，获取排序后序列的下标为 k 的元素作为输出即可。
与选择排序相似，每次选择待排序序列的最大元素加入已排序的末尾。

problem 15.5

问题 1 归约到问题 2 对于问题 1 的输入集合 S 直接设为传入问题 2 的输入集合即可，并令问题 2 的输入参数 $k = \frac{n+1}{2}$，输出即为集合 S 的中位数。

问题 2 归约到问题 1 不妨设 n 为奇数，阶为 k 的元素为 m
1. 当 k 等于 $\frac{n+1}{2}$ 时，将原集合传入问题 1，即可找到 m；
2. 当 k 小于 $\frac{n+1}{2}$ 时，遍历原集合 S，记录其元素最小值为 min，对于原集合 S，有 $n-k$ 个元素大于 m，$k-1$ 个元素小于 m。
3. 开辟新集合 temp，将前 $n-2k+1$ 个元素值赋为 min-1，并将 S 的 n 个元素放入其后。则集合 temp 中有 $n-k$ 个元素大于 m，$n-k$ 个元素小于 m。因此 m 为 temp 中位数，将 temp 传入问题 1，即可找到 m。
4. 当 k 大于 $\frac{n+1}{2}$ 时，同理记录其元素最大值为 max，开辟数组 temp，在数组 temp 中前 $2k-n-1$ 个赋值为 max+1。同样使得 m 变为 temp 中位数，将 temp 传入问题 1，即可找到 m。

Chapter 16

problem 16.2

1)

对于判定算法，选择图 G 中的 k 个点，共有 $C_n^k = \frac{n!}{(n-k)!k!}$ 种选择，选择 k 个点后，判断 k 个点之间是否均有边相连，则最多遍历 $C_k^2 = \frac{k(k-1)}{2}$ 条边。算法最多需要判断所有可能选择是否成立，因此时间复杂度为 $T(n) = \frac{n!}{(n-k)!k!} \times \frac{k(k-1)}{2}$，同时 k 为常数，则 $T(n) = O(n^k)$，即为多项式复杂度。

```
FakeCliqueJudge(G,k)
Loop:
    for each G_k in G:   /*选取G中的每个有k个点的子图*/
        for v in G_k:
            for w in G_k:
                if v==w: continue;
                if vw not in G: goto Loop;
                /*如果图G中不存在vw边，则判断下一个k点子图*/
        return true; /*每一对点均有边，则说明为团*/
    return false;
```

FakeCliqueJudge.func

2)

不能证明。因为题中要求的判定问题中，是根据给定的常数 k，判断是否具有 k 大小的团。而 $CLIQUE$ 问题的判定问题中，输入参数 k 是不确定的变量，在确定判定问题复杂度时，k 为常数的条件不再成立。伪最大团问题与 $CLIQUE$ 问题并不等价，因此并不能证明 $P = NP$。

problem 16.3

1)

对于 DNF-SAT，每个子句间以逻辑或相连，因此若使整个表达式成立，则使得其中任意一个子句成立即可。对于任意子句，其中各变量以逻辑与相连，只要其中不同时存在某一变量和该变量的非，则就可以对子句中每一布尔变量进行相应的赋值，使其为 $true$ 即可，就能使得子句为 $true$。如果每一子句都包含了布尔变量和布尔变量的非，则 DNF-SAT 中每一子句均不能为 $true$，最终结果为 $false$。显然在 $O(n)$ 时间内即可完成，即多项时间内可解，DNF-SAT 为 P 问题。

2)

CNF-SAT 等价地转换为 DNF-SAT 时，这个归约过程无法保证在多项式时间内完成，所以后续推理不成立。则不能得到 $CNF-SAT \leq_p DNF-SAT$，不能证明 $P = NP$。

problem 16.4

稠密子图为 NP 问题 对于任意猜测的解，（k 个点的形式）我们可以在 $O(k^2)$ 的时间内验证，这个子图中的边数是否至少有 y 条边，即解可以多项式时间内验证，所以是 NP 问题。

稠密子图为 NP 难问题 可通过证明最大团问题可以在多项式时间下归约到稠密子图问题，来进行证明。对于最大团问题其输入 G，k 直接作为稠密子图的输入 G、k，并令 $y = C_k^2 = \frac{k(k-1)}{2}$，显然归约为多项式时间。

输出的正确性： 对于一个子图 H，有 k 个顶点，则其最多有 C_k^2 条边，即完全图的情况。如果对于上述归约后的输入，稠密子图输出为 $true$，则说明 G 中存在 k 大小的完全图 (团)；稠密子图输出为 $false$，则说明 G 中任意 k 个顶点的子图，其边数均不足 C_k^2，即不存在 k 大小的完全图。

稠密子图为 NP 完成问题 综上最大团问题可以多项式时间归约到稠密子图问题，因为最大团问题是 NP 完全问题，并且稠密子图是 NP 问题，所以稠密子图是 NP 完全问题。

problem 16.5

SET-COVER 为 NP 问题 给定 k 个子集，我们可以在 O(kn) 时间内验证它们的并集是否是 U，所以我们可以在多项式时间内，验证这个问题的一个解，所以这个问题是 NP 的。

SET-COVER 为 NP 难问题 可通过证明 DOMINATION-SET 问题可以在多项式时间下归约到 SET-COVER 问题，来证明。对于 DOMINATION-SET 问题其输入 G，k，以 k 作为 SET-COVER 的输入 k，以图 G 的顶点集合作为输入 U，每个子集 S_i 为图 G 第 i 个点以及该点的邻居。显然归约为多项式时间。

输出的正确性： 如果对于上述归约后的输入，输出为 $true$，则说明存在大小为 k 的集合 $S_1', S_2', \cdots, S_k'$ 覆盖 U，而 S_i' 则对应了图 G 中的点 v_i'，即存在大小为 k 的支配集；如果输出为 $false$，同理，说明不存在大小为 k 的支配集。

SET-COVER 为 NP 完成问题 综上 DOMINATION-SET 问题可以多项式时间归约到 SET-COVER 问题，因为 DOMINATION-SET 问题是 NP 完全问题，并且 SET-COVER 是 NP 问题，所以稠 SET-COVER 问题是 NP 完全问题。