hw7 刘佳璇 231220105

第二十章

20.1 四个问题

1）

CLIQUE

优化问题：给定无向图 G ，求 G 中最大团的大小。

判定问题：给定无向图 G ，问 G 中是否存在大小为 k 的团。

KNAPSACK

优化问题：给定 n 个物品，大小为 $s[1\dots n]$ ，价值为 $c[1\dots n]$ ，求在背包中装物品的大小之和不超过 C 时价值之和最大是多少。

判定问题：给定 n 个物品，大小为 $s[1\dots n]$ ，价值为 $c[1\dots n]$ ，问是否有一种在背包中装物品的方案，满足物品的大小之和不超过 C ，且价值之和不超过 k 。

INDEPENDENT-SET

优化问题：给定无向图 G ，求 G 中最大独立集的大小。

判定问题：给定无向图 G ，问 G 中是否存在大小为 k 的独立集。

VERTEX-COVER

优化问题：给定无向图 G ，求 G 中最大点覆盖的大小。

判定问题：给定无向图 G ，问 G 中是否存在大小为 k 的点覆盖。

2）

充分性：假设优化问题的解为 x ，比较 x 与 k 的大小即可得到判定问题的解。

必要性：假设判定问题的解的时间复杂度为 $f(n)$ ，图的大小为 n ，则令 $k=0,\dots,n$ 依次求解即可得到优化问题的解，时间复杂度为 $nf(n)$ ，由多项式运算的封闭性可得优化问题依然为多项式时间可解。

3）

CLIQUE

验证一个大小为 k 的点集是否为团，只需要两两检验点之间是否有边，时间复杂度为 $\mathrm{O}(k^2)=\mathrm{O}(n^2)$ ，所以是NP问题。

KNAPSACK

给定若干个物品，验证背包中是否能全部装下且价值之和不超过 k 的物品，只需要检验所有物品的大小之和不超过 C ，价值之和不超过 k ，时间复杂度为 $\mathrm{O}(n)$ ，所以是NP问题。

INDEPENDENT-SET

验证一个大小为 k 的点集是否为独立集，只需要两两检验点之间是否有边，时间复杂度为 $\mathrm{O}(k^2)=\mathrm{O}(n^2)$ ，所以是NP问题。

VERTEX-COVER

验证一个大小为 k 的点集是否为点覆盖，只需要检验图中的每一条边是否与点集中的点关联，时间复杂度为 $\mathrm{O}(|E|)=\mathrm{O}(n^2)$ ，所以是NP问题。

20.2 证明多项式时间规约关系是传递关系

证明:

如果 $P \le_P Q, Q \le_P R$ ， $T(x)$ 是问题 P 到 Q 的归约，代价为 $f(n)$ ， $S(x)$ 是问题 Q 到 R 的归约，代价为 $g(n)$ ，那么 $S(T(x))$ 是问题 P 到 R 归约，代价为 $g(f(n))$ ，是多项式时间，故 $P \le_P R$ 。

20.3 解决 A 问题所需时间

$\mathrm{O}(n^2+n^4)=\mathrm{O}(n^4)$

20.4 “排序”与“选择”

排序到选择的归约:

排序算法调用 n 次选择算法，依次选出最小到最大的的元素。

选择到排序的归约:

选择算法调用排序算法，然后在已排序的序列中按下标或遍历选择元素。

20.5 中位数与阶为 k 的数

问题 1 到问题 2 :

问题 2 输入 $k=\dfrac{n}{2}$ 。

问题 2 到问题 1 :

先调用中位数算法，然后进行 $partition$ ，对左右两个子序列递归进行上述操作，直到选出的中位数是阶为 k 的数。

第二十一章

21.1 P、NP、NPC

1）

如果一个NPC问题 P 可以在多项式时间解决，那么 $\forall X \in NP, X \le_P P$ ，即任意一个NP问题都可以在多项式时间解决。

2）

如果一个NPC问题 P 不可以在多项式时间解决，那么对任意NPC问题 Q ， $P \in NP, P \le_P Q$ ，所以 Q 不可以在多项式时间解决。

21.3 伪最大团问题

1）

从所有点中选择大小为 k 的点集，共有 $\mathrm{O}(\binom{n}{k}) = \mathrm{O}(\frac{n!}{k!(n-k)!}) = \mathrm{O}(\frac{n^n}{k^k(n-k)^{n-k}}) = \mathrm{O}(n^k)$ 种选法，然后验证这个子集是否为团，时间复杂度为 $\mathrm{O}(k^2)$ 。所以算法总的时间复杂度为 $\mathrm{O}(n^k k^2)$ 。

2）

不能证明。因为最大团问题中的 k 是依赖于 n 的参数，而本题中 k 是常数。

21.4 DNF-SAT问题

1）

TODO

2）

该推理中未证明CNF-SAT问题的输入归约成等价的DNF形式需要多项式时间。

21.5 背包问题

1）

证明：

对于背包问题，调用子集和问题的算法求解元素和为 $i(i = k, \ldots, \sum_{i=1}^{n} c_i)$ 的子集，需要重复 $\mathrm{O}(n)$ 次；对每个所求的子集合， $\mathrm{O}(n)$ 检验大小之和是否超过 C 。归约的代价为 $\mathrm{O}(n^2)$ ，得证。

2）

令 $dp[i][j](0 \le i \le n, 0 \le j \le C)$ 为前 i 个物品装进容量为 j 的背包的最大价值，状态转移方程为

$$dp[i][j] = \begin{cases} max \begin{cases} c_i + dp[i-1][j-s_i] \\ dp[i-1][j] \end{cases} & i > 0, j > 0 \\ 0, & i = 0 \lor j = 0 \end{cases}$$

算法先从上到下、再从左到右计算， $dp[n][C]$ 即为所求的解。

3）

共计算 nC 个子问题，每个子问题的时间复杂度为 $\mathrm{O}(1)$ ，因此总的时间复杂度为 $\mathrm{O}(nC)$ 。 但当 ￥C 足够大时，算法输入的规模为 $\log C$ ，因此时间复杂度为 $\mathrm{O}(n2^{\log C})$ ，不是多项式时间。

21.6 支配集问题和集合覆盖问题

从支配集问题到集合覆盖问题的归约如下：给定无向图G和全集U，全集U中的元素是顶点集合，每个顶点集合 $S[1 \ldots n]$ 表示G中顶点 $v[1 \ldots n]$ 及其邻居构成的子集。求出无向图G的支配集D，D中元素 $v[i_1 \ldots i_k]$ 对应的顶点集合 $S[i_1 \ldots i_k]$ 即为全集U的集合覆盖。因此集合覆盖问题是NP难问题。

以下证明集合覆盖问题是NP问题：给出一个大小为 k 的子集集合 T ，遍历 T 中子集的所以元素，检验是否包含全集U中的所有元素，所需代价为 $\mathrm{O}(|U|)$ ，是多项式时间。

21.10 回避路径问题

从哈密顿路径问题到回避路径问题的归约如下：构造有向图 G' ，顶点是 $s, t, Z_i (1 \leq i \leq k)$ ，边是图 G 中相应的边，构造代价为 $\mathrm{O}(|V|+|E|)$ 。图 G' 中的哈密顿路径即为图 G 中EPP问题的解。

以下证明EPP问题是NP问题：给定有向图 $G=(V, E)$ 和一条从 s 到 t 的路径，检验路径上的每个点属于哪个不重叠区域，时间复杂度为 $\mathrm{O}(|V|)$ ，是多项式时间。