高级算法设计与分析 Assignment 2

1. Write down the main steps of proving the NP-Completeness of a problem.

Answer for 1:

假设要证明问题Y是一个**NP-completeness**问题

1. 首先证明Y问题是一个**NP**问题;
2. 选择一个已知的**NP-completeness**问题X;
3. 将问题X多项式归约到问题Y，即证明$X \le_p Y$。（即说明了求解问题Y不比求解问题X简单）

2. Given a graph, a dominating set is a subset of vertices such that any vertex not in this set is adjacent to at least one vertex in this set. The dominating set problem is to check whether a given graph has a dominating set of size at most k.

   2.1 Prove that the dominating set problem is in NP.

   2.2 Prove that the dominating set problem is NP-hard.

Answer for 2.1:

假设给定的图为$G=(V,E)$，且$|V|=n$，$|E|=m$。

对于一个给定的点集$S(|S|=k)$，判断点集S是否是一个dominating set problem的解，算法如下:

对于点集$V-S$中的每个点，判断其邻居中是否有至少一个点在点集S中，G中每条边会被至多枚举$O(1)$次，因此复杂度为$O(n+m)$。

因此dominating set problem是**NP**问题。

Answer for 2.2:

vertex cover problem: 给定一个图$G=(V,E)$，问图中是否存在一个大小不超过k的点集S，使得图中的每一条边都至少有一个端点在点集S中?

已知 vertex cover problem 是一个 **NP-hard** 问题，我们将通过把 Vertex cover problem 归约到 dominating set problem（即 vertex cover problem$\le_p$dominating set problem）来证明 dominating set problem 也是一个 **NP-hard** 问题。

对于任意一个 vertex cover problem（以下简称 VCP）的实例$<G=(V,E),k>$，我们都构造一个对应的 dominating set problem（以下简称 DSP）的实例$<G'=(V',E'),k>$，其中:

- 将图G复制到图G'中（复制所有点和边）；
- 对于V中的每条边(u,v)，即点u和点v的连边，在图G'中构造$k+1$个新顶点，记为$uv_1,uv_2,\cdots,uv_{k+1}$，并将这$k+1$个点都分别与点$u$和点$v$相连，即在图$G'$中新增边$(u,uv_i)$和$(v,uv_i)$，$i=1,2,\cdots,k+1$。

接下来我们证明 VCP 的实例$< G=(V,E),k >$有解**当且仅当**对应的 DSP 的实例$< G'=(V',E'),k >$有解。

(⇒)证明若给定的 VCP 的实例有一个大小为k解Sol_1，则我们能在对应的 DSP 的实例中也能找到一个大小不超过k解Sol_2：

- 令$Sol_2=Sol_1$，由于图G'中包含原图图G，因此Sol_2是合法的解；
- 根据 VCP 的定义可知，图G中的每条边都至少有一个端点在Sol_1中，由此可知在图G'中的每条原图G的每条边都至少有一个端点在Sol_2中，因此原图G中的每个点要么在Sol_2中，要么与Sol_2中的点相连；
- 对于构造中新增的点，由于原图G中任意边(u,v)的两点u,v至少有一个在Sol_2中（假设是u在Sol_2中），因此新增的点$uv_1,uv_2,\cdots,uv_{k+1}$都与$u$相连。

(⇐)证明若给定的 DSP 的实例有一个大小为k解Sol_2，则我们能在对应的 VCP 的实例中也能找到一个大小不超过k解Sol_1：

- 若给定的 DSP 的实例的解Sol_2中包含了某个新增的点uv_i，同时又包含了点u,v中的至少一个（假设是u在Sol_2中）。由于点uv_i只能支配点u,v，不能支配到除它之外其他由边(u,v)新增的点uv_j已，而点u可以支配到点v以及所有由边(u,v)新增的点，因此我们可以将点uv_i从Sol_2中删除，得到一个更小的解Sol_2'。Sol_2'也是一个合法解；

- 若给定的 DSP 的实例的解Sol_2中包含了某个新增的点uv_i，但不包含点u,v。那么Sol_2一定包含了由边(u,v)新增的所有点$uv_1, uv_2, \cdots, uv_{k+1}$，否则存在某个点$uv_j$即不在$Sol_2$中，又不被解集中的点支配。由此可知$Sol_2$中至少包含了$k+1$个点，即$Sol_2$的大小至少为$k+1$，那么$Sol_2$就不是一个合法解；
- 由上述两条可知若一个给定的 DSP 的实例一个大小为k解Sol_2，那么Sol_2中要么只包含原图G中的点，要么可以变成一个更小的只包含原图G中的点的解Sol_2'；
- 令Sol_1=Sol_2或Sol_2'，由上述分析可知Sol_1是一个合法解。

证毕。

3. Prove that: if we can check whether a graph has a clique (a complete graph) of size k in polynomial time then we can also find a clique of size k in polynomial time.

Answer for 3:

对于给定的图$G=(V,E)$，找到大小为k的团的算法如下：

1. 找到图中的一个点u，使$G'=G\backslash\{u\}$中依旧存在大小为k的团。（可以通过枚举所有点$v\in G$，询问$G'=G\backslash\{v\}$中是否存在大小为k的团来完成）
2. 如果找得到上述的点u，则将u从G中删去，即$G\leftarrow G\backslash\{v\}$，然后回到第一步。
3. 如果找不到这样的点u（算法停止时），说明剩下的图即为一个大小为k的团。

4. A graph is called a 2-plex if each vertex in the graph is not adjacent to at most one other vertex. Prove that it is NP-complete to check whether an input graph has a sub graph of at least k vertices that is a 2-plex.

Answer for 4:

我们定义一个大小为k的2-plex是图点集的一个大小为k的子集P_k，点集P_k的导出子图中每一个点都至多与P_k中除自身外的一个点不相邻。

性质1：在一个大小为$2k$的2-plex $P_{2k}=(V,E)$中存在一个大小至少为k的clique。

证明：考虑$P_{2k}=(V,E)$的补图$\overline{P_{2k}}=(V,\overline{E})$，其中$\overline{E}=\{(u,v):(u,v)\notin E\}$。根据2-plex的定义，补图中每个点的度数至多为1，且至多有k条边。现对补图中的点集做一个划分(X,Y)，其中度数为0的点放入X中，对于一条边$(u,v)\in\overline{E}$，将u放入X，将v放入Y。这样得到的(X,Y)中，点集X在原图中任意两点之间都有边，点集Y亦然，因此X和Y在P_{2k}中均是clique。又由于$|\overline{E}|\le k$，因此$|Y|\le k$，那么有$|X|\ge k$，证毕。

证明2-plex在NP中:

给定图$G=(V,E)$中一个大小为k的点集P，我们可以枚举这个点集中每个点，检查该点是否只与P中除自身外的一个点不相邻，若每个点都是，那么给定的点集P是一个2-plex，否则不是。时间复杂度为$O(k+|E|)$，因此2-plex在NP中。

证明2-plex在NPC中:

k-clique problem：给一个图$G=(V,E)$，问图中是否存在一个大小为k的点集使得点集中的任意两点之间都有边。

已知k-clique是一个**NP-complete**问题，接下来将k-clique问题规约到2-plex问题。

给定k-clique问题中的一个实例$G=(V,E)$，构造一个2-plex问题的实例$G'=(V',E')$，其中

- $V'=V_1\cup V_2$，$V_1=\{u_1:u\in V\},V_2=\{u_2:u\in V\}$，即将每个点$u\in G$复制两份，每个点记为$u_1$和$u_2$，并且说$u_1$和$u_2$在$G$的对应点是$u$。
- $E'=E_1\cup E_2\cup E_3$，$E_1=\{(u_1,v_1):(u,v)\in G\}$，$E_2=\{(u_2,v_2):(u,v)\in G\}$，$E_3=\{(u_1,v_2):(u,v)\in G\}$，即若$G$中有一条边$(u,v)$，则在$V_1$中有一条边$(u_1,v_1)$，在$V_2$中有一条边$(u_2,v_2)$，在$V_1$和$V_2$之间有边$(u_1,v_2)$和$(v_1,u_2)$。

性质2：根据定义可得，在G'中的任意两个点u和v之间有边当且仅当它们在G中对应的两个点之间有边。

（⇒）证明若G中有一个大小为k的clique，则G'中有大小为$2k$的2-plex。

假设G的clique为K，那么在G'中对应的$K_1\cup K_2$是一个2-plex，理由如下：

首先对于K_1中任意两点间有边，K_2中任意两点间有边。

其次对于$u_1\in K_1$和$v_2\in K_2$，它们之间没边当且仅当u_1和v_2对应G中的同一个点（由性质2得）。

因此$K_1\cup K_2$中每个点都与$K_1\cup K_2$除自身外恰好一个点不相邻，是一个2-plex。

证毕。

（⇐）证明若G'中有大小为$2k$的2-plex，则G中有一个大小为k的clique。

根据性质1，我们能在大小为$2k$的2-plex找到一个大小至少为k的clique K，对于K中的任意两个点，它们之间有边当且仅当G中的对应点之间有边（性质2），因此这K个点中的任意两个点在G中的对应点不会是同一个点。那么这k个点在G中的对应点也构成一个clique，大小为k。

证毕。

5. In the multiway cut problem, we are given a undirected graph G=(V,E) and some special vertices in V (called terminals). The problem asks us to delete the minimum number of edges from the graph such that no pair of terminals is connected. Please give a 2-approximation algorithm for this problem.

Answer for 5:

假设输入的无向图为$G=(V,E)$，给定的特殊点集为$T=\{t_1,t_2,\ldots,t_m\}$。

对于点t_i，令边集C_i为将t_i和$T\backslash\{t_i\}$割开的最小割。

每个C_i可以用最大流算法在多项式时间内得到：新建一个点x_i连向所有的$T\backslash\{t_i\}$，求t_i到x_i的一个最大流，在残余网络上构造一个最小割即可。

现证明$S=C_1\cup C_2\cup\ldots C_m$是该问题的一个2倍近似解。

首先，S是该问题的一个可行解，因为C_i将t_i都与T中的其他点都割开了，所以$G'=(V,E-S)$中，T中的任意两点都不在一个连通块内。

其次，假设最优解为S^*，证明$|S|\le 2|S^*|$。

考虑在$G_{s^*}=(V,E-S^*)$中，假设t_i所在的连通块为D_i（每个t_i所在的连通块一定互不相同），连通块D_i和连通块D_j间在G中的边集为$E_{i,j}$（$E_{i,j}\in S^*$且$\bigcup_{1\le i<j\le m}E_{i,j}=S^*$）。我们有

$$\sum_{1\le i<j\le m}|E_{i,j}|=|S^*|$$

由于C_i是t_i到$T\backslash\{t_i\}$的最小割，而$\bigcup_{j\ne i}E_{i,j}$是t_i到$T\backslash\{t_i\}$的一个割，我们有：

$$|C_i|\le\sum_{j\ne i}|E_{i,j}|$$

$$|S|\le\sum_{1\le i\le m}|C_i|\le\sum_{1\le i\le m}\sum_{j\ne i}|E_{i,j}|$$

$$|S|\le\sum_{1\le i\le m}|C_i|\le 2\sum_{1\le i<j\le m}|E_{i,j}|\quad(|E_{i,j}|\text{和}|E_{j,i}|\text{各被计算了}1\text{次})$$

$$|S|\le\sum_{1\le i\le m}|C_i|\le 2|S^*|$$

证毕。