

# Week 15

## 35.3-2

---

可以将顶点覆盖问题归约为集合覆盖问题。

在顶点覆盖问题 $G$ 中，将每条边看作一个元素，与他连接的两个结点看作集合。所以，每个元素恰好被两个集合同时包含。而每个集合包含的元素，就是在 $G$ 中与它连接的边。

该归约过程显然是多项式时间完成的。

集合覆盖问题也是多项式时间可验证的。

## 35.3-3

---

1. 找到当前 $|S|$ 最大的集合 $S_{max}$  ,  $S_{max} \rightarrow F_{return}$
2. 结点集合 $X = X - S_{max}$
3. 找到所有满足条件的集合 $S_i : S_i \cap S_{max} \neq \phi$
4.  $S_i = S_i - S_{max}$

重复1-4，直到 $|X| = 0$ ，返回 $F_{return}$

其中，2、3、4步的时间复杂度为 $O(|S_{max}|)$ ，合理使用数据结构第1步的时间复杂度为 $O(|S_{max}|)$

所以运行时间为 $O(\sum_{s \in \mathcal{F}} |S|)$

## 35.5-5

---

将数据结构 $L_i$ 中的每个数字，替换为二元组 $(N, B)$ 。其中， $N$ 代表数字， $B$ 是一个二进制数长度为 $n$ ，第 $i$ 位记录第 $i$ 个数字是否包含（0为不包含，1为包含）。

对于算法APPROX-SUBSET-SUM，将第2行修改为： $L_0 = \langle (0, 0 \dots 0) \rangle$

第4行的 $L_{i-1} + x_i$ 修改为：每个 $L_i - 1$ 中的 $N$ 加上 $x_i$ ，同时 $B$ 的第 $i$ 为置为1

第5行的TRIM修改为用 $L_i$ 元素的 $N$ 判断是否应该删除

最后，返回的 $z^*$ 改为满足要求的二元组 $(N, B)$ ，其中 $N$ 为 $L_i \cdot N$ 中最大。 $B$ 记录了该子集，通过匹配可以给出

子集的元素。