第 11 章《算法能力的极限》习题

1. 简述：

（1） 问题变换（或问题化简）和复杂性归约的基本思想。

对于某些问题，直接通过计算模型建立时间下界很困难，

假如 A 问题下界是 $\Omega(f(n))$，我们希望对 B 问题建立下界 $\Omega(f(n))$，此时我们就可以将问题 A 变换为问题 B。A 的任意实例可转化为 B 的一个实例，任何求解 B 的算法都可以求解 A

（2） 求解问题下界的几类主要方法。

a. 平凡下界：对问题的输入中必须要处理的项进行计数；同时对必须要输出的项进行计数

b. 信息论法：根据算法必须处理的信息量来建立效率的下界

c. 敌手法

（3） P、NP 和 NP 完全的概念。

P 类问题是一类能够用（确定性）的算法在多项式的时间内求解的判定问题，也称为多项式类型

NP 类问题是一类可以用不确定多项式算法求解的判定问题，也称为不确定多项式类型

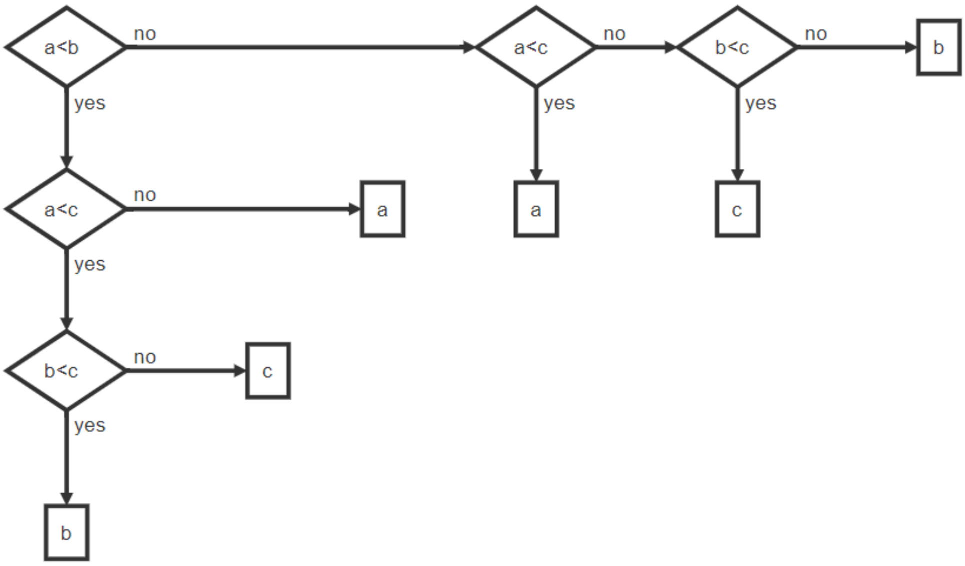
NP 完全：如果一个 NP 完全问题能在多项式时间内得到解决，那么 NP 类中的每个问题都能在多项式时间内求解，即 P=NP

（4） 证明问题 NP 完全性的基本思路。

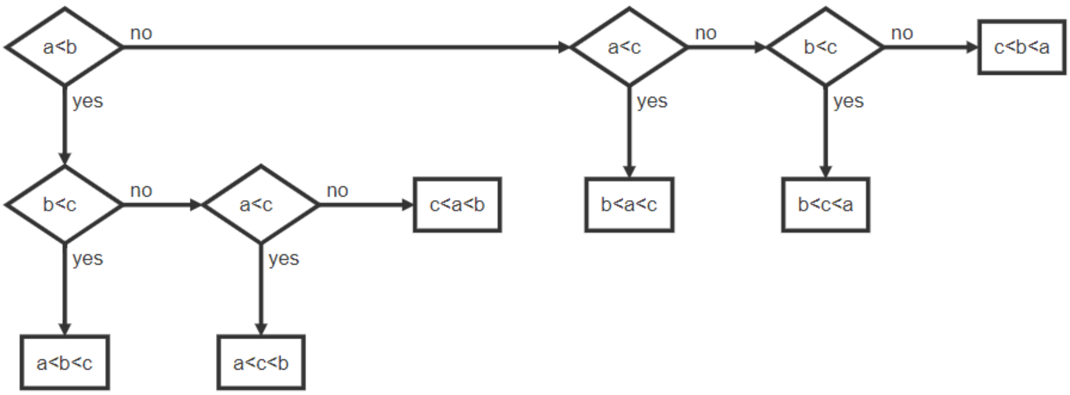
a. 存在一个 NP 完全问题 L

b. 对于$L_1 \in NP$，证明 L: $\propto_p L_1$

2. 考虑在三元素集合$\{a, b, c\}$中“求中值”问题，为求解该问题的算法画一棵决策树。



3. 对于“三元素的基本冒泡排序”问题，画出其决策树，并求出它们在最坏情况和平均情况下的键值比较次数。



最坏情况比较次数为 3 次；平均情况比较次数为$\frac{8}{3}$次

4. 定义下面每个问题的判定版本，并且简要描述问题的一个多项式时间的算法，它能够检验某个特定解是不是问题的一个解（可以假设一个特定解代表了检验算法的一个合法输入）。

（1） 背包问题

（2） 装箱问题