1. 二叉树(binary)

50分做法：

只需构建出二叉排序树并输出层次遍历，第二行随意输出yes或no。

100分做法：

按照二叉排序树的构建方法建树，并在层次遍历时给所有点标号。

若N为奇数，判断标号前(N-1)/2的点是否都有左右孩子，若都有，则是完全二叉树，否则不是；若N为偶数，判断标号前N/2-1的点是否都有左右孩子且标号N/2的点是否有左孩子，若都有，则是完全二叉树，否则不是。

1. 列车调度(manage)

30分做法：

直接搜索每一辆列车进入哪一条轨道。剪枝1：若当前列车进入空轨道，选择任意一个空轨道的结果是一样的。剪枝2：每一条轨道进入的列车的标号必须单调下降。

70分做法：

实际上就是求这个序列的最长上升子序列的长度。

简略证明：

假设最长上升子序列长度为K’，则其中任意两辆不会进入同一个轨道，因此K≥K’。至于K=K’可由数学归纳法详细证明。

用O(N²)的DP方法求最长上升子序列。

100分做法：

用二分法或树状数组(线段树)优化DP，时间复杂度O(NlogN)。

另：直接分析搜索过程，可以得到O(N²)的构造方法，加上一个二分就可以满分。（略）

1. 保留道路 (road)

10分做法：

任何暴力方法，例如枚举所有可能的maxG和maxS，然后判断连通性。

30分做法：

按照g属性从小到大排序，枚举maxG，对满足maxG的所有道路按照s属性从小到大排序，然后做kruskal，时间复杂度O(M^2logM)。

50分做法：

在30分基础上，发现每次只增加一条边，插入到上次的边集合中再做kruskal即可，时间复杂度O(M^2)。

100分做法：

依旧按照g属性从小到大排序。不断加入新边的过程中发现，当前的最小生成树只可能是由未加入新边的最小生成树的边和当前新边组成的共N条边中选出N-1条构成。因此维护一个最小生成树边集，每次只在N条边中做最小生成树，时间复杂度O(MN)。