T1

使用堆来模拟。我们定义多个面试官同时完成任务称为“事件”。

对于每个事件，存储完成任务面试官的集合。把接下来的几个人随意地分配给这些面试官。由此，我们可以确定小Z面试的时间，以及其中一个可以面试到小Z的面试官 x。

令事件集合为 E_i。

现在，考虑一下如果我们**按倒序遍历**这些事件会发生什么。

如果面试官 x 参与了其中一项活动（即 $x \in E_i$），那么任何与他同时完成的面试官都可能是贝茜的面试官。因此，每当贝茜可能的面试官和某一事件中的面试官相交时，该事件中的任何一个面试官最终都可能成为贝茜的采访者。

然后我们让答案的集合初始化为S={x}。

接着**倒序**枚举“事件”E_i：

可以用 bitset 实现。

时间复杂度 $\Theta(nlogk + n^2/w)$。~~并且跑得飞快。~~

T2

思路就是对于每一个字符串的子串都丢到一个哈希表去判断它是否在之前字符串中出现过，最后再把自己扔进哈希表。这样复杂度就可以降到大约为O（n size^ 2*k）(k为哈希的常数，**一般不大**) 要注意的是，为了保证对于每一个字符串而言，可能是它的子串的字符串都在前面被处理完了，所以要按字符串长度排序。

但是有一个细节，就是**如果一个字符串的子串有相同的话我们只计算一次**，但按照之前的思路会计算两遍，所以我们还需要一个小哈希去判断当前子串是否在**这个字符串**中出现过。但是问题又来了，对于每一个新的字符串而言我们都要清空前面的小哈希，这样的时间复杂度让人难以承受。解决方法很简单，用结构体哈希，多一个域标记这个子串是在第几个字符串中出现的即可。

T3

见PPT

T4

首先，我们知道，基环树是一棵树上加一条边所得到的。下文称这条边为特殊边，我们可以求出特殊边的两个端点 fm 和 to。

设 $f_{i,0/1/2/3}$ 表示 i 的子树在对应状态下的最小染色点数，对应状态如下：

- 0 表示点 i 未染色，且 i 的不包括父亲的邻接点都没有被染色（如果有点 i 在特殊边上，特殊边的另一点也为被染色）
- 1 表示点 i 未染色，且 i 的不包括父亲的邻接点中有一个点被染色
- 2 表示点 i 被染色，且 i 的不包括父亲的邻接点都没有被染色
- 3 表示点 i 被染色，且 i 的不包括父亲的邻接点中有一个点被染色

直接做树型 DP 即可。

对于特殊边上的两个端点 fm 和 to，枚举它们的颜色（枚举分别是否染色，只要枚举四次），在做树型 DP 时对这两个点特殊处理一下即可。

时间复杂度为 $O(n)$。