T1:

对于一个点，显然按照子树大小和权值和的比值升序来 dfs，确定一个起点时，树形 DP 显然。

对于一个起点 X，将起点变为 X 的某个儿子 Y 时的答案可以通过类似的办法维护出来。

时间复杂度 O($N\log N$)

T2:

先跑出一棵最小生成树，显然，对于一条已经在最小生成树中的边，答案为所有覆盖这条边的非树边的最小值，对于一条非树边，答案为覆盖的树边的最大值。

前者可以在树上打标记然后启发式合并得到，后者可以直接倍增求出。

时间复杂度 O(N $(\log N)^2$)

T3:

首先考虑所有行一起的答案。

设 f[x]为每一行在 x 之前的数字都被选走，最后连续 K 个为 x，且其他数字都没有连续出现 k 次的方案数。只考虑 x 连续出现 k 次的方案数是一个简单的组合数式子，考虑去掉别的数字出现 K 次的方案，不妨设第一次连续出现 K 次的数字为 y，则 y 对 x 的贡献也是 f[y]和一些组合数的乘积。将所有满足条件的 y 的贡献都减去即可。为了方便实现，我们在每一行最后加一个数字 N+1，则最后的答案就是$\frac{\mathrm{f[N+1]}}{K!}$。这样的单次复杂度是$N^2K$的，总复杂度为$N^2K^3$。

考虑固定行的起点，枚举行的终点，每次新加入一行，所有的组合数都可以维护，复杂度N^2K^2。

由于随机的性质，考虑满足条件的数对(x,y)，在只有一行的时候，数量为$\frac{N^2}{2}$，然后每多一行，原来的数对继续满足条件的概率都为$\frac{1}{2}$，所以数量/2，所以对于每个行起点，行终点的移动过程中，满足条件的数对(x,y)的总数为N^2，所以我们在新加入一行的时候维护每个 x 合法的 y 即可。

时间复杂度 O（$\mathrm{NK}(N+K)$）