**tree**：

令f(i,j,k)表示前i个点里选出了j个点，prufer序列长度为k的方案数。枚举每个点的出现次数转移即可。

f(i+1,j,k)+=f(i,j,k)

f(i+1,j+k,k+c)+=f(i,j,k)/c!

大小为s的答案为f(n,s,s-2)\*(s-2)!。

时间复杂度O(n^4)，可以用FFT优化至O(n^3logn)

**rab：**

每个点有3种情况：红胜，蓝胜，先手胜。树形dp求出每个点的情况。

考虑第二问，如果根是红胜的话，可以选任意一个叶子；如果是先手胜的话，需要将某个蓝胜的子树变成先手胜，或是将某个先手胜的子树变成红胜。

如果需要将蓝胜变为先手胜，同样考虑改变某个儿子，也有可能无解。

时间复杂度O(tn)

**hidden：**

只考虑向右的话，用f[S][i][j]表示当前已经处理了S中的条件，现在正在处理第i个条件且完成了前j个数，还需要多少个数。转移分两种情况：（1）填入第i个条件的下一个数，从f[S][i][j+1]转移；（2）枚举条件k使得只要接上k就一定能满足i，从f[S|(1<<i)][k][0]转移。

对于向左只要类似地加上两维[k][l]表示左侧正在处理第k个条件，完成了第lx[k]个数。转移时增加三种情况：（3）填入第k个条件的下一个数，从f[S][i][j][k][l-1]转移；（4）若第i个条件和第k个条件的下一个数相同，从f[S][i][j+1][k][l-1]转移；（5）若l=1，即第k个条件已经填完，枚举u，v使得第u个条件完成v个数以后接上k就能满足，从f[S|(1<<k)][i][j][u][v]转移。

特殊情况：i=0表示右侧尚未开始，k=0表示左侧已经结束。

注意填数时要保证不会违反另一个条件，即(1)填的数必须是条件k的前l个数，(3)填的数必须是条件i的前j个数。

由于转移顺序并不明显，需要用记忆化搜索。

时间复杂度O(2^n\*n^3\*10^2)