sol

by tsx

T1

- 枚举横着的切法。
- 接下来考虑对竖着的切法 dp，$f_{i,j}$ 表示前 i 列切了 j 刀时的答案。
- 转移直接枚举下一刀在哪里即可。
- 时间复杂度 $O(2^n n^3)$。
- 当然你闲着没事也可以枚举最终用了几刀，然后二分。

T2

- 一个重要的性质是说，假如我们记录了第一行删了 i 个元素，第二行删了 j 个元素，那么 S 里的元素是确定的。
- 我们可以直接将每个位置的 $|S|$ 的大小记录到一个棋盘上，每次可以往下/右走一格。
- 所以假如没有交换，直接记录到 $(1,1)$ 到 (i,j) 走过的格子的 $|S|$ 的 max 的最小值，然后直接转移即可。
- 考虑交换会影响什么。不难发现，交换 $i,i+1$ 实际上只影响第 i 行格子上的数，那么我们考虑再记录一个 (i,j) 到 (n,n) 的值，然后暴力更新第 i 行之后前后拼起来就好了。
- 时间复杂度 $O(nm)$。

T3

- 考虑状压 dp。
- 用 $f_{S,i}$ 表示当前有集合 S 中的颜色，目前树根在点 i 的答案， $g_{S,i}$ 表示当前有集合 S 中的颜色，树根在点 i，且 i 的度数为 1 的答案。
- 每次，我们可以通过 $f_{S,i}$ 再枚举一个 j 转移到 $g_{S\cup\{a_j\},j}$，而 $f_{S,i}$ 自身的转移可以通过枚举其中一个儿子合并得到。
- 为了防止重复，这里枚举其中一个儿子时，它的子树需要包含目前编号最小的颜色。
- 最后我们发现每个树被恰好算了 k 遍，答案除以 k 即可。
- 时间复杂度 $O(3^k n)$。

T4

- 首先 R 是无法删除的。
- 那么我们希望用 R 来把问题分成若干段。
- 注意到，我们每次删除 P 的时候，一定是删除的最开头的可以被删除的 P。
- 但 S 就没有这么好的性质了，所以我们的 dp 状态形如 $f_{i,j,k}$ 表示考虑前 i 个位置，第 i 个位置是 R，且第 i 个位置前面有 j 个 S 未被删除，后面有 k 个 S 要删除时的答案。
- 转移时，枚举下一个 R 的位置，那么中间的部分一定是 S 和 P 交替出现。
- 那一次肯定没法直接转移到，所以可以考虑再记录另一个 dp 为 $g_{i,j,k,l,0/1,0/1}$，其中 i,j,k 与上面类似，l 表示中间的部分里 P 还有几个，中间的部分的开头/结尾是 S 还是 P。

T4

- 看上去这东西不太能转移，因为考虑 ...RSPSPSR... 在删除 P 之后会变成 ...RSSPSR...，而这不再满足中间的 P 和 S 是交替的。
- 但是可以注意到 ...RSSPSR... 与 ...SRSPSR 是等效的，也就是说，若开头有两个 S，我们将第一个 S 和 R 交换是无关紧要的。
- 那么现在就可以转移了，注意到 f 的状态数有三维，转移是 $O(n)$，g 的状态数有四维，转移是 $O(1)$，所以时间复杂度为 $O(n^4)$。
- 注意 g 数组是可以滚动的，空间复杂度为 $O(n^3)$。
- 常数很小，可以通过。

T4

- 还可以更快一点吗？
- 注意我们刚才枚举的下一个 R 其实没有特别大的用处，所以我们可以记录 $h_{i,j,k,0/1/2}$ 表示当前消除到 i，且第 i 个位置前面有 j 个 S 未被删除，后面有 k 个 S 要删除，当前位置为 R/P/S 时的答案。
- 转移也是类似的。
- 时间复杂度 $O(n^3)$。
- 不过这个做法细节可能会更多一些。