20250311题解

A

开个循环，判断i是不是3的倍数然后输出就行了。

B

O(n^2)枚举就行了，注意判断编号。

C

直接开一个循环记录每种颜色的最小值，再记录最大的那个就可以了。注意豆子编号可能很大，建议用map。

D

这是一个广搜，但是怎么搜还是有讲究的。

考虑如果当前格有药水，且喝完药水能量比之前高，那么就应该喝掉。毕竟不会返回原来的格子。

bfs 时将标记有没有走过改为记录走到当前格子的最大的剩余能量。如果走到的格子记录的能量比现在的大就舍去，否则更新记录的能量并入队。可以使用优先队列将剩余能量较大的排在前面进行优化，时间复杂度多一个log，可以接受。

E

这就是洛谷P2986（甚至是简化版，去掉了边权），经典换根DP。

先设结点1为根结点，size[i]为此时以i为根的子树内的点权和，g[i]为从i的子树的所有点到i的距离距离总和，则f(1)=g[1]。

接下来换根，将根结点从x变成某个儿子y时，以y为根的子树的距离减少d(x,y)，其他点的距离增加d(x,y)，直接统计即可。

F

暴力O(n^2m)大概能拿38分？

但是用一个bitset优化就切了。因为求两个序列里相等的数字可以直接将两个序列异或一下然后找0的个数，而异或操作就可以使用bitset来优化。那么我们先把每个数变成二进制，bitset 维护每一位的情况，再将每一位的异或结果再或一次，即每一位异或结果都为 0 才能为 0 。统计最后序列中 0 的个数是否为奇数即可。时间复杂度O(n^2m\*log(W)/w)。

G

若单独考虑一个k的问题，解法是将(A[i],B[i])按照B[i]递增的顺序排序（之后的下标均为排序后的下标），接着枚举B[j]作为maxB[i]，要取出A[1...j]中最大的k个组成当前最大值。建立n棵主席树，每次查询A[1...j]的前k大，在第j棵主席树上二分。离散化版本是O(nlogn)。动态开点代码稍短，O(nlogV)，V为值域。

接下来按 k 从小到大的顺序考虑问题。假设 k=x 时选择 B[j]作为最大值是最优的。k=x+1 时，j 这个决策点由于比 j−1 多了一个可选数 A[j]，因此肯定增长幅度会比 j−1 更大一点，加上本来 j 就已经比 j−1 优秀了，j−1 就更不行了，归纳可得 1~j−1 都已经不需要再考虑了。

于是用分治来利用这一性质。设当前区间为 [l,r]，可选决策点为 [L,R]。取 [l,r] 的中间值 mid，枚举 [L,R] 中的决策点，获得 k=mid 时的最优决策点 P，则左边区间的可选决策点变为 [L,P]，右边区间可选决策点为 [P,R]，这样类似线段树得递归下去，每层的决策点区间长度和为 O(n)，故每层的时间复杂度为 O(nlogV)，共 logn 层，总时间复杂度 O(nlognlogV)。