solution

A.

考虑倒序进行维护，此时当你删除一个点时，这个点的贡献已经被之前的操作计算完毕。将此时的权值计入答案数组中即可。

需要实现子树加与单点查询，可以使用树状数组与差分进行维护，复杂度 $O(n\log n)$ 。

没有对其他常数（可能）更大的做法进行卡常。

B.

考虑 dp ，设 dp_i 为 dp 到 i 时的答案，枚举放的状态；

1. 直接放入上一个所在的管道。若 $x_i=x_{i-1}$ ，则 $dp_i=dp_{i-1}+b_{x_i}$ ，否则 $dp_i=dp_{i-1}+a_{x_i}$ 。
2. （考虑如何减少当次贡献）设 l_i 为上一个与 i 同色的出现位置。 $dp_i=dp_{l_i}+s(l_i+1,r_{i-1})+b_{x_i}$ 。

$s(l,r)$ 表示将 $[l,r]$ 的中的球放入同一管道的代价之和。上式的实际意义是，将 l_i 放到一个管道中，在此之后 $[l_i+1,i-1]$ 的球全部放入另一个管道中的答案。

然后你写了这个发现过不了样例！！！ 这里的原因是 l_i+1 的贡献可能被错误的计算了，l_i+1 未必会产生 a_{l_i+1} 的贡献。

考虑 dp_{l_i+1} 的意义，实际上可以使用 dp_{l_i+1} 进行转移，$dp_i=dp_{l_i+1}+s(l_i+2,r_{i-1})+b_{x_i}$ 。 dp_{l_i+1} 的值实际上为考虑上 l_{l_i+1} 的值后的答案，而且 l_i 的贡献也与 dp_{l_i} 部分相同，所以这样是对的。需要特判 $l_i=i-1$ 。

用前缀和对 $s(l,r)$ 进行维护即可。复杂度 $O(n)$ 。$O(n\log n)$ 的线段树做法可以通过 80 分。

C.

显然若 H 的数量为奇数，答案为 0 。不妨设其为 x 。

否则设在横方向设置了 u 道栅栏，纵方向设置 v 道栅栏，则一定满足 $uv=\dfrac{x}{2}$ 。

此时枚举 u,v ，最多只有 $d(\dfrac{x}{2})$ 种取值，由于还有 $1\le u\le n,1\le v\le m$ 的限制，实际情况要更少。

对于确定的 u,v ，考虑此时行的划分是几乎确定的（除了没有 H 的列可以在一定区间内自由选择），对列考虑，此时也可以像行一样进行维护，考虑每一个列与行的分界点，检查其中 H 的数量是否正确。单次的复杂度为 $O(mu)$ 。总复杂度不超过 $nm\log\log n$ ，并且很难卡满，可以通过。

D.

显然按照 dfs 序进行遍历最优。

考虑进行树形 dp ，设 $dp_{u,0}$ 为走完 u 的子树，老头初始在 u 上，且需要回去的答案，$dp_{u,1}$ 为不需要回去的答案。定义 w_u 表示不使用老头对 u 的子树进行标记的答案。显然，$w_u=2siz_u-d_u$，其中 d_u 为 u 子树的最大深度。

设出发点固定为 x ，我们只需要求 $dp_{x,1}$ 。

$$dp_{u,0}=\sum_{v\in S_u}\min(w_u,dp_{v,0}+2)$$

$$dp_{u,1}=\sum_{v\in S_u}\min(w_u,dp_{v,0}+2)-\max_{v\in S_u}(\min(w_u,dp_{v,0}+2)-\min(w_u,dp_{v,1}+1))$$ 。

对于每个儿子 v 而言，有两种选择：不移动老头，使用 w_u 的代价进行遍历；

移动老头，且还需要把老头移动回来，使用 $dp_{u,0}+2$ 的代价进行遍历。

对于 $dp_{u,1}$ 来说，可以选择一个点不把老头移动回来，所以代价减去 $\max(\min(w_u,dp_{u,0}+2)-\min(w_u,dp_{u,1}+1))$ 。

直接枚举 x 进行 dp 可以做到 $O(n^2)$ ，而使用换根 dp 可以做到 $O(n)$ 。可能卡了带 $\log$ 做法。

有实现细节，码量可能稍大。