Multi-University Training 01 – BUPT - Analysis

Problem A. Harvest Moon

枚举选择的种子并进行模拟。由于只能选择一种种子，模拟时每次选择最大的空地进行播种即可。需要稍微注意的是，如果从左上角开始种植 3 * 3 的完整方格，种完之后应当首先考虑右下角的 3 * 3 方格，它拥有当前非完整 3 * 3 的最大的面积，随后再考虑最右和最下的带状面积。

Problem B. Letter Tree

考虑整棵树上从根节点出发到每个叶节点所形成的字符串 S_i。对于某个节点 u 的询问，最优策略是在以他为根的子树内，找到深度满足条件且字典序最大的叶子 v，并沿着从 u 到 v 这条路径往下走即可（这些叶子的 S_i 的最长公共前缀必然不小于 Depth_u）。因此可以对树重构出 Trie 并得到每个节点在 Trie 中的映射，并由此通过一遍 dfs 得到每个节点的 S_i 的 rank。随后我们可以把原树按深度分为若干个数组，每层内按照 dfs 的时间戳排序。我们可以得到，对于 u 的在某一层的子孙，在该层中为连续的部分。所以我们每次询问(u,m)，我们只要找到深度为 Depth_u+m 层中 u 的子孙那部分。通过二分可以直接得到。我们可以通过线段树查询最值，由于题中并没修改操作，所以我们可以直接用 SparseTable 每次查询为 O(1)的复杂度。几种写法都是在时限内可以接受的。

Problem C. Partition

我们可以特判出 n <= k 的情况。
对于 1 <= k < n，我们可以等效为 n 个点排成一列，并取出其中的连续 k 个点。下面分两种情况考虑：
第一种情况，被选出的不包含端点，那么有(n – k – 1)种情况完成上述操作，剩下未被圈的点之间还有(n – k – 2)个位置，可以在每个位置断开，所以共 2^(n−k−2) ∗ (n−k−1)种方法。
第二种情况，即被选出的包含端点，那么有 2 种情况，并且剩余共(n – k – 1)个位置，所以共 2 ∗ 2^(n – k – 1)种方法。
总计 2 ∗ 2^(n – k – 1) + 2^(n – k – 2) ∗ (n – k – 1) = (n – k + 3) * 2^(n – k – 2)。

Problem D. Color the Tree

假设双方的起始点分别在 A 和 B。双方的最优策略是尽量往对方的方向行进，尽早占领对方的地盘，然后某个时刻在 AB 路径的中点 O 会合。此时以 O 点为割点，树被分成了若干个连通块，每个连通块的边权之和加起来作为这个连通块的价值。在会合后，双方会尽量抢价值最大的连通块，然后回来继续接着抢（只需要占领从 O 到某个连通块的边，该连通块显然就属于自己了）。因此对 O 点相连的所有连通块按照价值进行降序排序，除去双方之前已经走过的连通块，剩下的奇偶分组，当前的先手取奇数部分。但题目有不少 trick：

1. 双方一开始在同一个点（双方会合前不占领其他连通块）
2. 双方一开始相邻（双方会合前只占领一个连通块）
3. 双方会合前占领的连通块在排序后的序列中可能位于两端（序列被分成 1,2,3 段均有可能）

尽管题目的策略不难，但代码的细节处理需要比较谨慎。

Problem E. Deque

考虑题目的一个简化版本：使双端队列单调上升。对于序列 A 和队列 Q，找到队列中最早出现的数字A_x，则A_x将 Q 分成的两个部分分别是原序列中以A_x开始的最长上升和最长下降序列，答案即为这两者之和的最大值。而对于本题，由于存在相同元素，所以只要找到以A_x为起点的最长不下降序列和最长不上升序列的和，然后减去两个里面出现A_x次数的最小值即可。

Problem F. Magic Ball Game

对于某个询问(u, x)，首先判断从根到 u 的路径上是否有点权等于 x。若存在，则到达当前点的概率是 0。
不妨定义往左走的路径为“左路径”，往右走的路径称为“右路径”。设左路径上大于 x 的点权有a_1个，小于 x 的点权有a_2个，则通过所有左路径并到达节点 u 的概率是$p_1 = \left(\frac{1}{2}\right)^{a_1} * \left(\frac{1}{8}\right)^{a_2}$。
类似地，设右路径上大于 x 的点权有b_1个，小于 x 的点权有b_2个，则通过所有右路径并到达点 u 的概率$p_2 = \left(\frac{1}{2}\right)^{b_1} * \left(\frac{7}{8}\right)^{b_2}$。最后的答案就是$p = p_1 * p_2$.
求根到 u 的路径上大于（小于/等于）x 的点权数量，可以在离线之后对树做一次 dfs，并用树状数组（线段树，平衡树）维护当前路径中的所有点权。总复杂度是$O(Qlog_N)$.

Problem G. Occupying the Cities

把平面上所有可以沿直线到达的的点连边（与所给的线段判相交），然后求一遍任意两点间的最短路。二分答案，对可以一步到达的城市进行连边，随后的问题就转为一个经典的最小路径覆盖了。

Problem H.Park Visit

首先如果 k 小于等于直径长度，那么答案为$k - 1$。
如果 k 大于直径长度，设直径长度为 r，那么答案为$r - 1 + (k - r) * 2$。

Problem I. 3 idiots

记录 A_i 为长度为 i 的树枝的数量，并让 A 对它本身做 FFT，得到任意选两个树枝能得到的各个和的数量。枚举第三边，计算出所有两边之和大于第三条边的方案数，并把前两条边包含最长边的情况减掉就是答案。希望你们不会对这个题的做法感到无趣。

Problem J. I-number

可以证明，所求出来的答案 y 一定会有$y \leq x + 20$. 因此暴力累加然后判断即可。

Problem K. Cards

由于卡片上的数字不大，可以直接暴力分解因数然后判断是否满足各个条件。对于第四个条件，不难发现 x 的因数之积最后一定是x^y的形式，因此只需判断 y 的奇偶。
完成第一问之后，我们会得到$2^4 = 16$种不同的卡片（满足了不同的条件）。随后的做法就非

常多了，较为简单的一种是直接2^{16}枚举取哪几种卡片，取的时候按照分数从高到低拿前 K 张就可以。如果以当前满足了哪些条件作为状态进行类似多重背包的 dp 的话，由于卡片的数量比较多，还需要进行二进制拆位处理，写起来就不如第一种做法简洁了。