牛客竞赛
AC.NOWCODER.COM

2024 牛客
暑期多校训练营

MagicalFlower

HR.NOWCODER.COM

A - 玲

题目大意

- 有 n 堆石子，第 i堆大小为 a_i。有一个石子是特殊的，每次可以在一堆中选择一个子集去检测是否存在特殊石子。
- 你要找到特殊石子，最小化最坏情况需要的操作次数。
- q 次区间询问。
- $n, q \leq 5 \times 10^4, m \leq 1000$

题解

- 第一次检测到之后，直接二分即可。设当前大小为 x，则需要 $\lceil \log_2 x \rceil$ 次操作。
- 每次肯定在最大的 a_i 里取石子。
- 一个贪心：每次取最大的一整堆石子，如果在这堆里就直接二分，否则继续去选第二大的那堆。
- hack：$n = 2, a = [1, 3]$。最优策略是在 3 的堆里选 2，如果在 2 里花一次确认，否则剩下 $[1, 1]$ 也能一次确定。

题解

- 令贪出来的答案是 ans'，考虑判定答案是否可以为 $ans'-1$。
- 贪心过程中第一次因为取了整堆导致答案会达到 ans' 的时刻。此时不能取整堆，只能取一个 $2^{\lceil \log_2 a_i \rceil -1}$，并留下大小为 $a_i - 2^{\lceil \log_2 a_i \rceil -1}$的一堆。
- 为了优化上述过程，把所有 $\lceil \log_2 a_i \rceil$一样的堆一起处理。
- 对于所有 $\lceil \log_2 a_i \rceil = k$ 的堆，除了 a_i最小的那个都没有取 2^{k-1}的意义。因此可以快速处理掉其他的，求出最小值后判断是否需要取 2^{k-1}。若需要则把 $a_i - 2^{k-1}$放到 $\lceil \log_2 (a_i - 2^{k-1}) \rceil$ 这些堆里一起处理。

题解

- 一些写代码需要注意的事情和细节。
- 预处理出前 i 个二进位第 j 堆的相对大小。$rank_i$ 可以由 $rank_{i-1}$ 线性转移来。
- 对于每个 k，预处理出所有 $\lceil \log_2 a_i \rceil = k$ 的堆的区间的 $\min$。k 个序列总长是 n, 可以用 st 表处理。
- 为了查询能 $O(1)$ 找到端点，预处理出 $cnt_{i,j} = \sum_{k \le j} [\lceil \log_2 a_k \rceil == i]$。
- 需要特判一下 $a_i = 0$ 和 $a_i = 2^k$ 的情况，因为 2^k 会比同一类的多一位。
- 时间复杂度 $O((n+q)m)$。

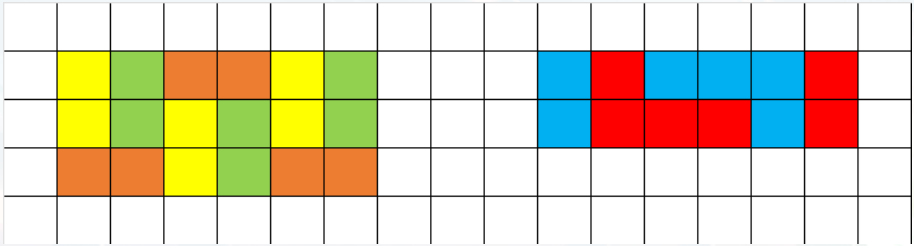
B - 珑

题目大意

- 要用 1×2 覆盖 $n \times m$。
- 有短边短边能否相邻，长边长边能否相邻两种限制。两种限制是否出现四种情况下，要求出是否存在覆盖方案。
- 多测。
- $n, m \leq 10^9, 0 \leq a, b \leq 1$。

题解

- $nm \bmod 2 = 1$ 无解。
- $n = 1, m = 2$ 或 $n = 2, m = 1$ 有解。
- $a = 0, b = 1$ 时，考虑构造 $3 \times 2k$ 和 $2 \times 2k$如下。因此就能叠出来 $n \times 2k, (n > 1)$。排除掉前面情况，只有 $1 \times 2k$构造不出来，容易证明此时无解。



题解

- $a=1,b=0$ 时，直接连成一排可以得到 $1\times 2k$。
- 简单分类讨论可以发现其他情况无解。
- 结合 $a=1,b=0$ 和 $a=0,b=1$ ，可以得到 $a=b=1$ 时只要 $nm \bmod 2=0$ 就有解。
- $a=b=0$ 只有 $n=1,m=2$ 或 $n=2,m=1$ 有解。

C - 骰

题目大意

- 完形填空有 n个题 m个选项（n是 m的倍数），已知答案是配平的。
- 平时的你做对 k 个题的概率是 P_k，且对于每种做出 k 个题的情况概率是相等的。
- 今天你做完之后发现自己的结果也是配平的，求你的分数期望 $\mathrm{mod}998244353$。
- $m \leq n \leq 2000, 0 < P_i < 998244353$。

题解

- 考虑求出所有配平的答案里，做对 k个题的方案数 f_k。
- 所有做对 k个题的情况里，配平的概率为 g_k，则有 $g_k = \frac{f_k}{\binom{n}{i}(m-1)^{n-i}}$
- 利用 g_k和 P_k 可以用贝叶斯写出答案 $ans = \frac{\sum g_i P_i i}{\sum g_i P_i}$。

题解

- 不妨设答案为 $aaabbbcccddd\cdots$。
- 计算 f_k 可以考虑容斥，先计算出钦定了 k个位置是正确的方案数，再进行一次二项式反演即可。
- $dp_{i,j}$ 表示考虑了前 i个选项（也就是前 $\frac{in}{m}$个题），钦定了 j个题是正确的。
- 转移枚举第 i个选项有 c_i个题是正确的，其余的 $\frac{n}{m}-c_i$个题要扔到最后随便选。
- 最后随便选的组合数是 $\binom{n-\sum c_i}{\frac{n}{m}-c_1,\frac{n}{m}-c_2,\cdots,\frac{n}{m}-c_m}$，可以把 $\frac{1}{(\frac{n}{m}-c_i)!}$的系数扔到 dp 里计算。
- 状态数 $O(nm)$，转移 $O(\frac{n}{m})$，总时间复杂度 $O(n^2)$。

D - 子

题目大意

- 有一个字符串 S和空串 T，每次把 S一个字符丢到 T里任意位置。
- 最小化过程中的编辑距离之和，把这个权值称为 $f(S)$
- 对所有长度为 n字符集为 m且 $f(S) \leq limit$的字符串，求权值之和。
- 即 $\sum_{S=\{1,2,\cdots,m\}^n, f(S)\leq lim} f(S) \bmod 10^9+7$。
- $n, m \leq 10^7$。

题解

- 先求一个字符串的权值。
- 当 $|S| = i, |T| = n - i$时，可以写出一个显然的上界 $\max(i, n-i) - \min(i, n-i, match)$，其中 $match$ 是把相同字符匹配起来的最多匹配数。
- 因为至少要把长度弄成相同的，其次剩余 $\min(i, n-i)$对字符里最多只有 $match$对字符能做到对位相等。
- 事实上这个上界也可以达到。先求出一组匹配，按照左端点从小到大的顺序，把右端点扔到 T的末尾。然后把剩余字符随便匹配，按照左端点在 S里的顺序，把右端点插入到 T里的对应的位置。最后随便把仍然在 S 里的元素插入到 T 即可。

题解

- 注意到 $match = \frac{n-\sum_{i=1}^{m}(cnt_i \bmod 2)}{2}$，其中 cnt_i是数字 i出现的次数。
- 令 $k = \sum_{i=1}^{m}(cnt_i \bmod 2)$，则对权值的限制转化为对 k的限制。
- 把 $\sum_{i=0}^{n} \max(i, n-i) - \min(i, n-i, match)$ 写成一个关于 k 的二次函数 $A + Bk + Ck^2$。
- 枚举 k，写出生成函数

  $$ans = \sum_{i=0}^{lim}(A + Bi + Ci^2)\left[\frac{x^n}{n!}\right]\binom{m}{i}\left(\frac{e^x - e^{-x}}{2}\right)^i\left(\frac{e^x + e^{-x}}{2}\right)^{m-i}$$

- 令 $f(x) = \frac{1}{x^{m/2}}\sum_{i=0}^{lim}(A + Bi + Ci^2)\binom{m}{i}\left(\frac{x-1}{2}\right)^i\left(\frac{x+1}{2}\right)^{m-i}$，则 $ans = \frac{1}{2^m}\left[\frac{x^n}{n!}\right]f(e^{2x})$。

题解

对 $f(x)$求导可以得到

$$f'(x)=\sum_{i\le lim}(A+Bi+Ci^2)\binom{m}{i}$$
$$(i(x-1)^{i-1}(x+1)^{m-i}+(m-i)(x-1)^i(x+1)^{m-i-1})$$
$$=m\sum_{i\le lim}(A+Bi+Ci^2)\binom{m-1}{i-1}(x-1)^{i-1}(x+1)^{m-i}$$
$$+m\sum_{i\le lim}(A+Bi+Ci^2)\binom{m-1}{i}(x-1)^i(x+1)^{m-i-1}$$

题解

考虑求出 $mf(x)-xf'(x)$。

$$\begin{aligned} mf(x)-xf'(x) =& m\sum_{i=0}^{lim-1}\binom{m-1}{i}(x-1)^i(x+1)^{m-i-1}(-B-(2i+1)C) \\ &+ m\binom{m-1}{lim}(A+Blim+Clim^2)(x-1)^{lim}(x+1)^{m-lim-1} \end{aligned}$$

前者递归到一个子问题，考虑如何求后者，即求 $(x-1)^k(x+1)^{m-k}$。

题解

继续对 $g(x)=(x-1)^k(x+1)^{m-k}$ 求导。

$$\begin{aligned} g'(x) =& k(x-1)^{k-1}(x+1)^{m-k}+(m-k)(x-1)^k+(x+1)^{m-k-1} \\ =& \frac{kg(x)}{(x-1)}+\frac{(m-k)g(x)}{x+1} \end{aligned}$$

两侧对齐一下系数即可线性求出。

于是 $f(x)$ 可以 $O(m)$ 求出。这里认为 n,m 同阶，为了线性算答案，需要线性筛一下 i^n。总时间复杂度 $O(n)$。

E - 安

题目大意

- 同时进行 n 场战斗，第 i场两方的血量为 a_i, b_i，前者属于 Alice 后者属于 Bob。
- Alice 和 Bob 轮流操作，每次可以选择一场比赛将对手的血量减少 1，若减少到 0 则获得了本场游戏的胜利。
- 总计 n 场游戏，双方都想赢得最多的游戏，求最后 Alice 能赢多少场。
- $1 \leq n \leq 10^5, 1 \leq a_i, b_i \leq 10^9$。

题解

- 若初始情况下 $a_i \neq b_i$。则 Alice 可以采取如下策略：
- 我只赢所有 $a_i > b_i$ 的位置。
- 每当 Bob 把 a_i 的血量打到和 b_i相等时，我下一轮立刻打 b_i一滴血。
- 这样始终保持 $a_i > b_i$，最后一定是 b_i先到 0。
- 同理 Bob 也可以采取同样的策略赢下所有 $b_i > a_i$ 的位置。
- 两个策略夹一下可以发现两个策略均达到最优，答案即为 $\sum[a_i < b_i]$。

题解

- 若存在 $a_i = b_i$，则双方谁先操作到 i 谁赢。
- 在有 $a_i = b_i$ 的情况下，操作外面一定不优。因为对方可以在外面应一手，相当于两人空过一回合。
- 因此假设有 k 个位置 $a_i = b_i$，则 Alice 可以赢 $\lceil \frac{k}{2} \rceil$ 场。
- 时间复杂度 $O(n)$.

F - 红

题目大意

- 给定 n, k，构造一个长度为 n、字符集为 $\{0, 1, 2\}$ 且恰好有 k 个平方子串的字符串，或者判断无解。
- $n, k \leq 10^5$

题解

- 首先考虑 $k=0$ 怎么做。
- 一个构造是 $a_i = paarity(i) - parity(i-1) + 1$，其中 $parity(i)$ 表示 i 在二进制下 1 的数量的奇偶性。
- 注意到这个序列是无限长的。
- 证明可以参考这个。
- 事实上这一步做法非常多。我随便写个搜，$n = 10^6$ 跑 $0.3s$。也有一个迭代的做法。

题解

- 令 f_i 表示连续 i 个 a有多少平方串。
- 一个想法是：如果把 $k=0$的构造中的一个字符扩展成很长的一段。那么一个长度为 $l_1+l_2+\cdots+l_k$的字符串，就有 $\sum_{i=1}^{k} f_{l_i}$ 个纯色的平方串。
- 事实上我们可以证明，按照上述方法构造出来的字符串仅包含纯色平方串。这个证明就是说 $S=T$的必要条件是把相同一段缩一起后也相等，然后分类讨论一下即可。

题解

- 然后我们注意到，令 $l_{k+1}=1$ 就可以得到一个比刚刚构造的字符串长度多 1 且平方串个数不变的字符串。
- 因此我们只需要求出 g_i 表示想构造出 i 个平方串，串长最小是多少。
- 求 g_i 只需要写个简单的背包 dp 即可，时间复杂度 $O(k\sqrt{k})$。
- 但我们刚刚构造不出来的 (n,k) 二元组，就真的没有解吗？

题解

- 如果你熟知 runs 理论，应该可以分类讨论一下证明。
- 首先需要证明 runs 之间不存在嵌套关系的字符串一定不优，这个讨论一下边界的重合部分即可。
- 其次我们可以不妨假设整个串是一个 runs ，并且其嵌套了子 runs。此时平方串个数最多的字符在形如 $aaaabaaaab$ 时取到。
- 根据我们算出来的 dp 数字可以逐个考虑每个 k，其对应最大的构造不出来的 n，即便使用形如 $aaaabaaaab$的字符串也达不到 k 平方串。
- 因此暴力验证了其在 $n, k \leq 10^5$内是正确的，由于两者高阶项的常数有区别，在较小处暴力验证后，可以证明对于任意 n, k均正确。

G - 豆

题目大意

- 一棵以 1 为根 n 个点的有根树，按照排列 P的顺序依次执行 lct 中的 access 操作，得到了某棵树 T。
- 你需要求出有多少排列 P 依次执行 access 后会得到 T。
- 对 T 有 q 次修改，每次对 T 执行 access(u) 操作。
- $1 \leq n, q \leq 2 \times 10^5$。

题解

- 对于一条实链，显然链底是最后操作的，其他的点之间顺序无所谓。
- 如果把实链缩成点，用链底的操作代表整个点的操作，则每个点必须比其父亲点早操作。
- 上面两条性质是充分必要的，因此可以建出如下的树：
- 实链底向此实链的父亲实链底连边。
- 非实链底向实链底连边。
- 则每个点的出现时间要比他父亲的出现时间早。

题解

- 问题转化为给定一棵树，要求在点上写一个排列，使得每个点比其父亲的点小。
- 这是一个经典问题，答案是 $\frac{n!}{\prod_{i=1}^{n} size_i}$。
- 可以用概率的方法理解，也可以直接树形 dp，根据 dp 式子归纳。
- 树的形态难以直接维护，考虑其在原树上的组合意义。令 S为实链顶的集合，可以发现答案即为 $\frac{n!}{\prod_{i \in S} size_i}$，其中 $size_i$ 表示节点 i在原树上的子树大小，这个不由修改而改变。
- 可以简单地暴力维护 S 集合，用树剖即可做到，当然用 lct 也可以，时间复杂度 $O(n \log n)$。

H - 入

题目大意

- 有一个无向图，每个点有点权 a_i且互不相同。
- 你在一个无向图上做梯度下降算法，每次会选周围点权最小的点走过去，如果比当前点权大就不走了，令行走过的点的数量称为这次梯度下降的时间。
- 你可以自由选择起点和 a_i，最大化梯度下降的时间。
- $1 \leq n \leq 40, 1 \leq m \leq \binom{n}{2}$

题解

- 称经过的点集合为 S，我们可以把 S以外的点权赋为充分大以保证不会走出去。
- 假设 S 里经过的顺序为 $a_1, a_2, \cdots, a_{|S|}$。则 (a_i, a_{i+1})之间一定有边，且对于 $i+1<j$需要满足 (a_i, a_j) 之间没边，否则可以从 i 直接跳到 j。
- 先枚举起点，再爆搜路径，同时维护哪些节点可以访问。每次新加入一个点就把上一个节点的邻居全都 ban 掉。

题解

- 写一发发现过了，考虑证明其时间复杂度。
- 假设当前路径的终止节点为 u 令 k 表示 u 有几个出度没有被 ban。
- 一旦往某个方向 dfs 之后，其余的 $k-1$个点一同被 ban 掉。因此时间复杂度可以写成 $T(n)=T(n-k)\times k$，在 $k=3$ 时取到最大，单次搜索时间 $T(n)=O(3^{n/3})$。
- 算上枚举起点的时间复杂度，总时间复杂度 $O(n3^{n/3})$。
- 单次的状态数是 $3^{n/3}$。但如果对于每个状态要枚举所有 n个可能合法或不合法的点，时间复杂度就会多个 n。如果只枚举合法的点，则就不会多这个 n。

I - 骨

题目大意

- 有一个联通带权无向图有 n 给点 m条边，边权在 $1 \sim k$之间。
- 对于一个生成树，令其边权可重集合中出现次数最多的数字为众数，若有多个众数则取最小的一个。你需要求出最小众数生成树。
- $1 \leq k \leq 10, 1 \leq \sum n, \sum m \leq 5 \times 10^4$

题解

- 考虑枚举答案 c，首先把所有颜色为 c 的边依次加入空图中，如果这条边合并了两个连通块，则称为成功的加入。设成功加入了 $limit$ 条边，则其他边的颜色数量必须小于等于 $limit$ 。
- 考虑枚举一个颜色集合 S，把所有颜色在 S 里的边依次加入，设有 cnt 条边成功加入了。如果 $cnt + limit * (k - |S|) < n - 1$ 则一定不合法，因为其他每种颜色的边最多加入 $limit$ 条，总数不到 $n - 1$ 条肯定不合法。
- 如果对于所有集合 S 都满足，我们认为 c 是合法的。

题解

- 上述做法为什么对呢？
- 本题还有个拟阵交做法，朴素实现可以做到 $O(n^2 \log n)$或 $O(n^2 k)$。
- 如果你会拟阵交，可以发现上述做法相当于枚举拟阵交算法的扩张停止状态。因此可以证明其与拟阵交算法等价，间接证明了其正确性。
- 可以对于不同的答案一起枚举集合 S。朴素实现的化，总时间复杂度 $O(2^k \sum m + Tk2^k)$。
- 再除个 k 应该也是可以的，更低的复杂度应该也有机会，不过不是本题考查重点。

J - 相

题目大意

- 给定 n, d, k。求有多少个有标号无根树满足 n 个点、每个点度数小于等于 d、且存在一条链包含 $1 \sim k$ 的节点（这条链可以包含其它节点）。
- $1 \leq d < n \leq 500, 1 \leq k \leq n \leq 500$。

题解

- 首先特判掉 $k=1$，此时可以把 k 改成 2。
- 首先把最小的包含 $1\sim k$ 的链拿出来。可以认为链上的节点编号从左到右为 $1\sim k$。最后给答案乘上 $\frac{k!}{2}$。
- 我们可以用如下结构描述一棵树。
- 建一棵大小为 a_1 的树且所有点的度数小于等于 d，且标记一个度数为 1且编号不为 1的点。
- 建 $k-2$棵大小为 $a_i(1<i<k)$ 的树且所有点的度数小于等于 d，且 1号点的度数小于等于 $d-1$，且标记一个度数为 1且编号不为 1的点。
- 建一棵大小为 a_k 的树且所有点的度数小于等于 d，且 1号点的度数小于等于 $d-1$。

题解

- 前 $k-1$ 棵树的那个一度点表示下一棵树的 1 号点，这样就会连起来一棵大树。
- 第 i 棵树的 1 号点对应了拼起来的树的 i 号点。
- 因此要求 $\sum_{i=1}^{k} a_i = n + (k-1)$。
- 令 f_i, g_i, h_i 分别三种树有 i 个节点时的数量。
- 则做一个简单 dp 即可求出答案。

题解

- f_i, g_i, h_i 的求法是类似的。这里只介绍 g_i的求法。
- 考虑 prufer 序列，可以发现编号为 i 的点度数为 d_i的有标号树的数量为 $\binom{n-2}{d_1-1,d_2-1,\cdots,d_n-1}$。
- 因此可以设计一个 dp，$dp_{i,j}$ 表示考虑了标号的前 i个点，度数的总和为 j时的 $\prod \frac{1}{(d_i-1)!}$ 之和。
- 总时间复杂度 $O(n^3)$。分块 ntt 可以做到 $O(n^2)$。多项式复合逆或其他多项式科技可以做到 $O(n\text{polylog}(n))$。

K - 思

题目大意

- Alice 和 Bob 在玩猜数游戏。
- Alice 在心中想了一个整数 x，Bob 每次可以向 Alice 询问一个整数 y。Alice 会告诉 Bob 是否 $x \geq y$。
- Alice 会在心中记录此次询问的代价，也就是 $|x-y|$，注意 Bob 并不会知道此次询问的代价是多少。
- Bob 已知 Alice 心中想的数字 x 在 a_i 中。Bob 为了确定 x 是多少可以向 Alice 进行若干次提问，他希望最小化最差情况下每次询问的代价之和。
- $1 \leq n \leq 2000, 1 \leq a_i \leq 10^9$。

题解

- 令 $dp_{l,r,c}$ 表示目前已知答案在 a_l 到 a_r 之间，且在 l 左侧的询问次数 - r 右侧的询问次数为 c，这里 c 有可能是负数。
- 每次询问的代价为 $|x-y|$，在得知询问结果后可以得到 y的系数并计入贡献，最后确定答案后计算 y的贡献，即 $dp_{i,i,c}=ca_i$。
- 转移可以枚举 $y\in(a_k,a_{k+1}]$，并 $O(1)$ 算出最优的 y 转移。

题解

- 为了优化上述算法有如下几个观察：
- c 的绝对值看起来不大。
- $dp_{l,r,c}$的决策点在 $dp_{l,r-1,c}$的决策点右侧。
- 由于转移权值可以写成一个单调下降函数的和单调上升函数的 $\max$，因此转移函数是单谷的。
- 于是可以固定 l, c后对 r和决策点双指针。
- 令 c为某个常数，写一发就过了。时间复杂度为 $O(n^2|c|)$。

题解

- 为什么 c 的绝对值不大呢？我们有如下几个观察：
- 令 $v = \max a_i$，则答案小于等 V。策略是直接二分，代价为 $(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots)v = v$。
- 若 $c = 0$ 时询问的 $y < \frac{1}{3}v$。则 x 在右侧的代价至少为 $\frac{2}{3}v$。x 在左侧时当前轮代价最多 $\frac{1}{3}v$，此后也最多 $\frac{1}{3}v$。因此 x一定在 $[\frac{1}{3}v, \frac{2}{3}v]$ 之间。
- 当 $c > 0$ 时只会赋一个 c更趋近于 0 的趋势，因此 $|c|$ 是 $O(\log v)$ 的级别。
- 事实上笔者猜测 $|c|$ 是 $O(\frac{\log v}{\log \log v})$ 级别。
- 时间复杂度 $O(n^2 \log n)$。

L - 知

题目大意

- 你已知第 i场比赛的获胜概率是 $\frac{a_i}{100}$。
- 你可以不停的执行如下操作：
- 选择第 $i+1$ 场胜率减少 $\frac{1}{100}$，并让第 i 胜率增加 $\frac{1}{100}$。
- 你希望最大化赢得所有比赛的概率。
- $1 \leq T, n, a_i \leq 100$。

题解

- 若 $a_i < a_{i+1}$，则给 a_i 增加 1，给 a_{i+1} 减少 1，则答案一定不会变劣。
- 写个调整上去发现过了。
- 精细实现可以做到 $O(Tn)$。

题解

- 为什么呢?
- 考虑其前缀和数组 sum_i，每次操作相当于对 sum_i做单点加 1。
- 考虑 $\frac{sum_i}{i}$ 的后缀最大值位置。因为 sum只能增加，因此这些位置一定不会减少。
- 已知平均分配乘积最大，因此我们构造的方法达到了答案的上界。
- 这个证明也是精细实现的一种方式。

M - 不知

题目大意

- 两个宝箱的其中一个有 c 金币，另一个什么都没有。
- 你可以花 1 金币问哪个宝箱里有金币，回答有 $p = \frac{a}{b}$的概率是对的，$1 - p$的概率是错的。
- 你可以问若干次之后打开一个宝箱，并获得里面的金币。
- 请你最大化获得金币的期望。
- $1 \leq c \leq 1000, 1 \leq a, b \leq 10$。

题解

- $p \in \{0, 0.5, 1\}$ 都很好做，对于剩余部分，不失一般性的，我们假设 $p \in (0.5, 1)$。
- 假设左侧和右侧的盒子得到的结果次数分别为 x, y，假设 $x \geq y$。则若宝石在左侧，得到该结果的概率为 $P = p^x(1-p)^y$，右侧的概率为 $Q = p^y(1-p)^x$。因此在左侧的概率为 $\frac{P}{P+Q}$，即为 $\frac{p^{x-y}}{p^{x-y}+(1-p)^{x-y}}$。
- 不难发现只和 $x-y$ 有关，即当 $|x-y| = i$ 时，你可以停止游戏，并期望获得 $c \times \frac{p^i}{p^i+(1-p)^i}$ 枚硬币。

题解

- 考虑令 $f(i)$ 表示 $x-y=i$ 到结束期望要操作几次，不难发现 $f(-n)=f(n)=0$。
- 对于其他 i，有 $f(i)=p\times f(i+1)+(1-p)\times f(i-1)+1$。考虑作差，$g(i)=f(i+1)-f(i)$。
- 则有 $g(i)=\frac{(1-p)g(i-1)-1}{p}$。为了方便起见，我们将下标整体平移，得 $f(0)=f(2n)=0$，求 $f(n)$。
- 考虑设 $g(0)=x$，令 $c=\frac{1-p}{p}$ 则有 $g(i)=c^ix-\frac{1}{p}\times\sum_{j=0}^{i-1}c^i=c^ix-\frac{c^i-1}{p(c-1)}$。
- $f(2n)=f(0)+\sum_{i=0}^{2n-1}g(i)$，因此 $\sum_{i=0}^{2n-1}c^ix-\frac{c^i-1}{p(c-1)}=0$，即 $\frac{c^{2n}-1}{c-1}x=\frac{c^{2n}-2n(c-1)-1}{p(c-1)^2}$，最终得 $x=\frac{c^{2n}-2n(c-1)-1}{p(c-1)(c^{2n}-1)}$。

AC.

题解

- 带入 x，整理式子可以得到 $f(n)=\frac{n(c^n-1)}{p(c-1)(c^n+1)}=\frac{n((1-p)^n-p^n)}{(1-2p)((1-p)^n+p^n)}$。
- 由于直接问都有 $\frac{c}{2}$ 的收益，因此最终取的 n 是至多 $\frac{c}{2}$ 的。实际上 $c\le 1000$ 时只会 n 取到 16 以下，直接算就行了，并不需要高精。
- 不计算高精度，时间复杂度 $O(Tc)$。

THANKS!

AC.NOWCODER.COM