题解

糖果游戏

- 有 n 个瓶子，标号分别为 1,2,⋯,n。初始状态下第 i 个瓶子里有 ai 颗糖果，第 i 个瓶子的容量为 bi。每个瓶子有一个分裂系数，第 i 个瓶子的分裂系数为 ki。
- Alice 先手，轮到某一个玩家操作时，她可以选择一个瓶子 i，并选择一种操作：
- 将第 i 个瓶子的糖果数量变为原来的 ki 倍，即 $a_i \leftarrow a_i \cdot k_i$。
- 向第 i 个瓶子中添加一颗糖果，即 $a_i \leftarrow a_i+1$。
- 但是所有操作需要遵循如下规则：每次操作后糖果的数量不能超过该瓶子的容量 bi，否则该操作是不被允许的。
- 现在，给定所有瓶子的糖果数、容量以及分裂系数，你需要帮她们求出 Alice (先手) 必胜还是 Bob (后手) 必胜。

算法一

- 对于 $b_i < a_i \cdot k_i$ 的测试点，操作 1 无法执行。因此只有操作 2，只需判断 $\sum b_i - \sum a_i$ 的奇偶性即可。(奇先胜，偶先负) 期望得分 8 分。

算法二

- 容易发现到每个瓶子可以看成一个独立 ICG 游戏，因此只需要求出每个游戏的 SG 值即可。
- 由于一个状态至多只有两个后继状态，因此 SG 值只有可能是 0,1,2.
- 对瓶子数不超过 200 的情形，只需要对每个瓶子从大到小进行 DP 计算 SG 值 fi。
- 时间复杂度 O(∑bi)，期望得分 24 分。

算法三

- 对于 k=2 的情形：
- 若 b 是奇数，则容易证明所有奇数点必败，偶数点必胜。(只需注意到奇数点只能走到偶数点。)
- 对于偶数 2i，设 f2i 为它的 SG 值，则 2i 只能转移到 2i+1 (必败) 和 4i，归纳可知 f2i=3−f4i，于是 O(logbi) 计算即可。
- 若 b 是偶数，设 b=4q+r (r∈{0,2})。
- 则容易证明当 a≤q 时，游戏 (a,b) 和游戏 (a,q) 同胜负。
- 当 a>q 时至多两次操作 1，简单讨论即可。
- 时间复杂度 O(∑logbi)，结合上述算法期望得分 44 分。

算法四

- 不难发现上述结论可以直接推广到任意偶数 k=2c。
- 唯一注意的点是此时要设 b=c^2q+r，再进行递归。
- 当 k 为奇数时，结论会有变化。
- 注意到 a+1 和 ka 的奇偶性不同，因此可以证明 f⌊b/k⌋=2。
- 更一般地，可以构造数列 {xn} 满足
- x1=⌊b/k⌋,xn+1=⌊(xn−1)/k⌋(n∈N∗)
- 则 fi=2 当且仅当 i∈{xn}。
- 其余情况直接 O(logb) 处理即可。
- 时间复杂度 O(∑logbi)，期望得分 100 分。
- 注意如果使用乘法不要爆 long long，适当的地方可以使用除法代替。

英雄联盟

- 他会在一开始设置一个参数 x% 。如果第一刀不暴，则第二刀的暴率增加到初始值的 2 倍;如果还是不暴，就继续增加到初始值的3 倍，以此类推，当叠加到 100% 以上的时候，默认下一发一定暴击。
- 当一次触发了暴击，暴击概率又会重置为x 。
- 求暴击率。

算法一

- 我们可以先算出来期望多少刀暴击一刀，然后输出1/ans就是答案。
- 期望多少刀可以通过枚举到第i(0<=i<100/x)刀还没有暴击的概率相加，就是期望多少刀暴击。
- 到第i刀还没有暴击的概率是(1-x)*(1-2x)*...*(1-ix)
- 暴力算就行了。

质数

- 给定一个复数序列，支持区间乘，区间赋值，区间询问素数个数。

算法一

- 我们假定如果 x 是素数，那么 -x 也是素数，询问则是问正素数个数。
- 容易发现的是，除了 1 外的非素数在赋值前都不会变成素数了。
- 使用平衡树维护序列中的连续段，遇到区间乘法就搜索，如果子树中都不是素数或 1 就跳过打标记。
- 这里可能会遇到问题，一个数被乘好几次还是素数或一怎么办？
- 容易发现乘的数必须是 0, 1, -1，这些全部打标记解决。
- 维护的信息只有子树正素数个数，负素数个数，是否存在素数，或 1。容易解决。
- 容易发现会被统计进入答案的素数不会超过值域。

算法二

- 现在是复数了，可能会产生奇异的状况。
- 例如：(2 + i)(2 - i) = 5，他变成了素数。
- 对复数稍微有点了解的同学会知道，复数的乘法就是模长相乘，辐角相加，也就是说这里模长是素数平方，他就可能是个素数。
- 使用类似刚刚的做法，如果一个数的模长平方因子个数超过 3，则它必定不是素数。而乘 0, -1, 1, i, -i 外的数，模长必定增加。
- 维护子树中是否存在因子个数不超过 3 的数，以及 辐角为 0°,90°,180°,270° 的模长是素数的数的个数，容易打标记解决。

隔膜

- 有一个n*n的棋盘，初始有一些棋子。每次选择一个k*k的没有棋子的正方形，然后任意选一个位置（可以不在k*k内）放一个棋子。双方轮流操作，没法操作输。

算法一

- 看到博弈题，显然要开始找一个队友玩一下了。
- 首先，如果一开始没法找到一个k*k的全空正方形，那么肯定是输出yc
- 其次，如果小c可以通过放一个棋子，使得小l没法找到一个k*k的全空正方形，那么输出rx
- 否则初始的时候至少有两个不相交的k*k的正方形。
- 当一个人操作后，不可能从存在两个不相交的正方形变成没有正方形。
- 而如果一次操作后所有正方形相交，那对手肯定赢了。
- 所以当最后只剩下两个不相交的正方形的时候。操作的那个人输了。
- 所以我们只要判断剩下的空格子个奇偶性即可，奇数小c赢，偶数小l赢。
- 我们只要判断剩下的空格子个奇偶性即可，奇数小c赢，偶数小l赢。