练习地址

https://www.nowcoder.com/acm/contest/202#description

出题人期望的难度

- 前期题
  - 卡牌游戏，平衡二叉树，矩阵乘法，数据排序
- 中期题
  - 数格点，生命游戏，排队，魔法阵
- 后期题
  - 字符串的幂，数组合并，游戏

卡牌游戏

$$\sum_{i=0}^{K-1} \frac{n-i}{m-i}$$

平衡二叉树

$$f(h) = \begin{cases} h & h \le d \\ f(h-1) + f(h-d-1) & h > d \end{cases}$$

高度为 n 的平衡二叉树，不平衡度最大时，必然是一个高度为 n-1 的满二叉树和一个高度为 n-d-1 的最小平衡二叉树，最小平衡二叉树的大小可以通过上面的公式求出，答案就是 2^{n-1} - 1 - f(n-d-1)

矩阵乘法

- 注意到，矩阵 B 是一个二进制矩阵。考虑红色的部分，它将和 B 的若干列向量点乘。若我们将红色部分的长度切块，例如分成 8 个 bit 一组，B 的列向量就只有 256 种情况

- 将这些值预处理，就可以用取值代替乘法操作了。

数据排序

- 注意到 $n \le 15$，可以用 $O(3^n)$ 或者 $O(3^n)$ 的DP求解。
- $f[S]$ 表示集合 S 是前 $|S|$ 大的元素，然后枚举剩下的元素的子集即可。

数格点

- 这道题跟皮克定理没有任何关系。
- 将多边形分拆成上下凸壳，然后做梯形分解，问题可以化简为一个线段下符合要求的点的个数。
- 化简后的问题可以写成标准的类欧几里得的形式。

生命游戏

- 考虑将生命游戏的坐标转45度，变成往 (1,1), (1,-1), (-1, 1), (-1,-1) 四个方向移动，这样 X 和 Y 就独立了。
- 很容易证明，单个细胞在第 n 时刻会有 $4^{bits(n)}$ 个存活，且这些坐标可以用形如 $(\pm 2^k, \pm 2^k)$ 的向量组合表示出来。这里的 k 为 n 的二进制中 1 的那些位。
- 一开始有两个生命的时候，在第 n 时刻他们可能会有部分重叠，求出重叠的部分容斥即可。

生命游戏

- 可以将其中一个作为参照系，统计有多少种方案，另一个细胞通过两组形如 $(\pm 2^k, \pm 2^k)$ 的向量移动能到达这个细胞的位置。这一步枚举 n 然后从低位到高位暴力就好。
- 考虑到 n 可能很大，可以分开枚举 n 的高位和低位，然后再组合到一起。

排队

首先考虑 $g_{i,j}$ 表示 i 个人在 j 时间段内，0 时刻有人来，且 i个人都进行了等待的期望与方案数

枚举 j 时刻来了 k 个人，则：

- 方案数 $gc_{i,j} = gc_{i,j-1} + gc_{i-k,j-1} \times \binom{i}{k}$
- 期望 $ge_{i,j} = ge_{i,j-1} + (ge_{i-k,j-1} + k$ 个人的等待时间 $\times gc_{i-k,j-1}) \times \binom{i}{k}$

然后 $f_{i,j}$ 表示 i 个人在 j 时间段内，原问题的期望与方案数

枚举一开始的 k 个人都进行了等待：

- 方案数 $fc_{i,j} = fc_{i,j-1} + gc_{k,j} \times fc_{i-k,j-kd} \times \binom{i}{k}$
- 期望 $fe_{i,j} = fe_{i,j-1} + (ge_{k,j} \times fc_{i-k,j-kd} + fe_{i-k,j-k*d} \times gc_{k,j}) \times \binom{i}{k}$

需要额外考虑 $k*d > j$ 时，$fc_{i,j}$ 加上 $gc_{i,j}$，$fe_{i,j}$ 加上 $ge_{i,j}$

最终答案为 $fe_{N,M}/fc_{N,M}/N$，其中 $fc_{N,M}$ 应该等于 $(M+1)^N$

复杂度 $O(N^2 \times M)$

魔法阵

考虑三个点最终的偏移向量分别为 $\vec{V}_1, \vec{V}_2, \vec{V}_3$。

则 $P_1 + \vec{V}_1$ 围绕 $P_2 + \vec{V}_2$ 旋转 60° 后与 $P_3 + \vec{V}_3$ 重合。

设旋转向量 $\vec{r} = (\cos 60^\circ, \sin 60^\circ)$

即 $P_2 + \vec{V}_2 + (P_1 + \vec{V}_1 - P_2 - \vec{V}_2) * \vec{r} = P_3 + \vec{V}_3$

得到 $P_2 + (P_1 - P_2) * \vec{r} - P_3 = \vec{V}_3 - (\vec{V}_2 + (\vec{V}_1 - \vec{V}_2) * \vec{r})$

$P_2 + (P_1 - P_2) * \vec{r} - P_3$ 表示将 P_1 围绕 P_2 旋转 60° 后与 P_3 的距离，记为 D

将向量视为围绕着原点的点，则原问题转化为

- 找一个最小的半径 R，使得圆内存在三点 V_1, V_2, V_3 满足 $V_3 - (V_2 + (V_1 - V_2) * \vec{r}) = D$

魔法阵

$V_2 + (V_1 - V_2) * \vec{r}$ 表示以 V_1, V_2 为顶点的正三角形的另一个顶点

对于一个半径为 R 的圆，我们希望 V_3 到上述点的距离最远。

显然三个点都必须在圆周上，建立函数表达式可以求出，最远距离为 $3R$。

故上述为题答案为 $\frac{D}{3}$。

注意枚举一开始旋转的方向，可能为 $-60°$。

字符串的幂

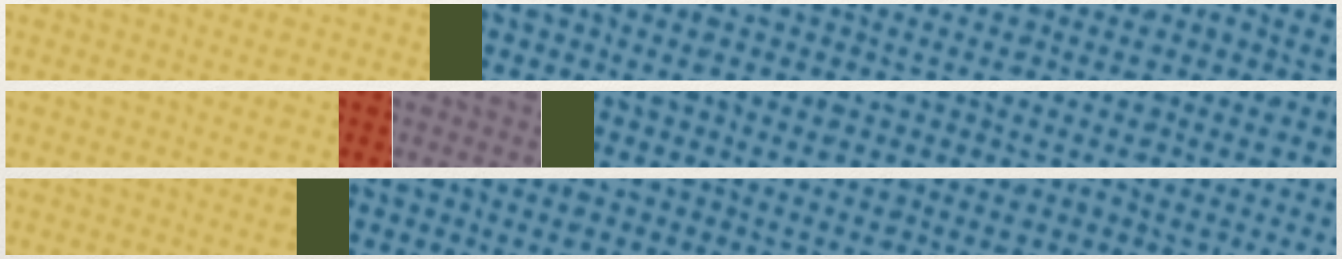
- s = a ^ n, n = 1 一定是一个答案。
- ∣a∣=1, ∣a∣=2, n = 2 需要单独计算。
- ∣a∣≥ 3, n ≥ 3 的时候，枚举 ∣a∣，则有三种情况
  - 前 ∣a∣ 个字符组成的子串是 a
  - 后 ∣a∣ 个字符组成的子串是 a
  - 中间 ∣a∣· (n-2) 个字符组成的串是 a ^ (n-2)，它可能出现的位置最多只有三种，可以枚举。

字符串的幂

- 枚举确定了 a 之后，依次删去 S 前缀的 a 和后缀的 a
- 此时有两种情况：
  - 剩下的串与 a 或空串的编辑距离为2
  - 剩下的串最前面 |a|±1 的串，后面 |a|±1 的串与 a 的编辑距离为 1
- 暴力枚举上面的所有情况，用哈希判断字符串相等即可。

数组合并

- 对每个询问，考虑合并后的数组中前 k 个数的最大值：



- 假设最大值 x 在红色的位置，深绿色的位置是每个数组中第一个大于 x 的位置，那么必然是黄色区域优先于红色，优先于紫色，优先于深绿色和蓝色。
- 此时只需要二分这个最大值即可快速求出第 k 个数所在的位置，这一步可以用线段树上二分来完成。

游戏

- 基本思想：分块
- 将所有人按照度数分为小点和大点：小点的度数不超过 sqrt(M)，大点的数量不超过 sqrt(M)。

游戏

- 我们需要为每一个大点维护一个当前在线的好友列表：
  - 当一个小点上线时，更新邻接大点的在线列表
  - 若邻接的大点有在线的，则与第一个组队
  - 当一个小点和一个大点组队时，将小点标记为大点的小弟，并从所有大点的在线列表中将其删除
  - 之后有人想和该小点组队，则直接通知其大佬
  - 即使一个大佬下线，他的小弟仍在这个大佬的队伍中，直至该小弟下线（此时大佬成为虚结点）
  - 当大佬上线时，直接访问自己的在线好友列表，并把其中没有大佬的在线小点纳为小弟。

游戏

- w 这样当小点上线、下线时，所有与之相关的操作的总复杂度为 O(sqrt(M))。
- 而大点上线时，邀请别的大点复杂度为 O(sqrt(M)).
- 而由于只有没有大佬的在线小点才需要通知，所以通知小点的复杂度均摊是 O(N) 的。