WC模拟赛题解

King_George

2019.1.2.

石子游戏

题目大意：给 n 个数，选出最多的数异或和为 0。

石子游戏

吐槽。

石子游戏

算法一：$O(2^n)$ 枚举每个数选不选。

石子游戏

算法一：$O(2^n)$ 枚举每个数选不选。
算法二：$dp[i][x]$ 表示前 i 个数中选择的数异或和为 x 时，最多可以选择多少个数。转移 $O(A)$，总复杂度 $O(nA^2)$。

石子游戏

算法一：$O(2^n)$ 枚举每个数选不选。
算法二：$dp[i][x]$ 表示前 i 个数中选择的数异或和为 x 时，最多可以选择多少个数。转移 $O(A)$，总复杂度 $O(nA^2)$。
算法三：令 $X = \bigoplus_{i=1}^{n} a_i$，将问题转化成删去最少的数使得异或和为 X。考虑最短路，对每个点 u 向 v 连一条有向边，当且仅当存在一个 a_i 使得 $u \oplus a_i = v$。以 0 为起点跑 BFS 即可。复杂度 $O(A^2)$。

石子游戏

算法一：$O(2^n)$ 枚举每个数选不选。
算法二：$dp[i][x]$ 表示前 i 个数中选择的数异或和为 x 时，最多可以选择多少个数。转移 $O(A)$，总复杂度 $O(nA^2)$。
算法三：令 $X = \bigoplus_{i=1}^{n} a_i$，将问题转化成删去最少的数使得异或和为 X。考虑最短路，对每个点 u 向 v 连一条有向边，当且仅当存在一个 a_i 使得 $u \oplus a_i = v$。以 0 为起点跑 BFS 即可。复杂度 $O(A^2)$。
算法四：令 $f[i][x]$ 表示选 i 个数，异或和能否为 x。可以发现有用的 $f[i]$ 是 $log\ A$ 级别的。转移可以用FWT优化。复杂度 $O(Alog^2A)$，也可以做到 $O(AlogA)$。

车

题目大意：将 n 个车摆在 $n \times n$ 的棋盘上，每个格子最多摆放一个，并且每行每列和**两条最长的对角线**上至少有一个车，并且有 m 个格子不能摆放。问方案数。

车

吐槽。

车

算法一：$O(n!)$ 枚举。

车

算法一：$O(n!)$ 枚举。
算法二：$m = 0$ 找规律。

车

算法一：$O(n!)$ 枚举。
算法二：$m = 0$ 找规律。
算法三：容斥，问题变为每次强制一些位置有棋子，求方案数。

车

算法一：$O(n!)$ 枚举。
算法二：$m = 0$ 找规律。
算法三：容斥，问题变为每次强制一些位置有棋子，求方案数。再容斥一下，先不考虑对角线上有棋子的条件，算出行和列上都有棋子的方案数，然后强制一条对角线为空，减去这样的方案数，再加上两条对角线都为空的方案数。

车

算法一：$O(n!)$ 枚举。
算法二：$m = 0$ 找规律。
算法三：容斥，问题变为每次强制一些位置有棋子，求方案数。
再容斥一下，先不考虑对角线上有棋子的条件，算出行和列上都有棋子的方案数，然后强制一条对角线为空，减去这样的方案数，再加上两条对角线都为空的方案数。不考虑对角线上限制很简单，答案是 $k!$。

车一条对角线为空

考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。

车一条对角线为空

考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。

剩下行列的方案数为 $k!$。

车一条对角线为空

考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。
剩下行列的方案数为 $k!$。
强制对角线上一些格子有棋子就是组合数 $\binom{k}{i}$。

车两条对角线为空

仍然考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。

车两条对角线为空

仍然考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。
剩下行列的方案数为 $k!$。

车两条对角线为空

仍然考虑容斥。强制对角线上一些格子有棋子，然后计算剩下的行、列方案数。

剩下行列的方案数为 $k!$。

强制对角线上一些格子有棋子的方案数不再是组合数了。考虑 $(i,\ i)$, $(i,\ n-1-i)$, $(n-1-i,\ i)$, $(n-1-i,\ n-1-i)$ 这四个格子放了多少个棋子。可以对每个 $i < \frac{n}{2}$ 求出这四个格子放置 k 个棋子的方案数，然后背包就行了。时间复杂度 $O(2^m\ n^2)$，可以用FFT优化。

求和

题目大意：$\sum_{i=0}^{N} \sum_{j=0}^{M} \binom{i}{j} \cdot [i \equiv 0 \pmod 2] \cdot [j \equiv 0 \pmod 2] \bmod k$。

求和

吐槽。

求和

算法一：$O(nm)$ 枚举。

求和

算法一：$O(nm)$ 枚举。
算法二：分治fft或者多项式除法。

求和

算法一：$O(nm)$ 枚举。
算法二：分治fft或者多项式除法。
算法三：$\sum_{i=0}^{N}\sum_{j=0}^{M}\binom{i}{j}\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{i=0}^{N}\sum_{j=0}^{M}[x^j](x+1)^i\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{j=0}^{M}[j\equiv 0 \pmod 2]\cdot[x^j]\sum_{i=0}^{N}(x+1)^i\cdot[i\equiv 0 \pmod 2]$。

求和

算法一：$O(nm)$ 枚举。
算法二：分治fft或者多项式除法。
算法三：$\sum_{i=0}^{N}\sum_{j=0}^{M}\binom{i}{j}\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{i=0}^{N}\sum_{j=0}^{M}[x^j](x+1)^i\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{j=0}^{M}[j\equiv 0 \pmod 2]\cdot[x^j]\sum_{i=0}^{N}(x+1)^i\cdot[i\equiv 0 \pmod 2]$。将 N 变为偶数，令 $F(x)=\frac{(x+1)^{N+2}-1}{(x+1)^2-1}$，则所求即为 $\sum_{i=0}^{M}[i=0 \mod 2]\,[x^i]F(x)$。

求和

算法一：$O(nm)$ 枚举。
算法二：分治fft或者多项式除法。
算法三：$\sum_{i=0}^{N}\sum_{j=0}^{M}\binom{i}{j}\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{i=0}^{N}\sum_{j=0}^{M}[x^j](x+1)^i\cdot[i\equiv 0 \pmod 2]\cdot[j\equiv 0 \pmod 2] = \sum_{j=0}^{M}[j\equiv 0 \pmod 2]\cdot[x^j]\sum_{i=0}^{N}(x+1)^i\cdot[i\equiv 0 \pmod 2]$。将 N 变为偶数，令 $F(x)=\frac{(x+1)^{N+2}-1}{(x+1)^2-1}$，则所求即为 $\sum_{i=0}^{M}[i=0 \mod 2]\ [x^i]F(x)$。考虑如何做除法，令 $a_i=[x^i](x+1)^{N+2}$，$b_i=[x^i]F(x), K=s2^t$，$s=1 \mod 2$，如果分别求出 ans 对 s，2^t 取模的值就可以通过 CRT 合并来解决。

求和模 s 意义下的结果

由于 $gcd(s, 2) = 1$，所以在模 s 意义下存在 2 的逆元 2^{-1}，所以 $b_0 = a_1 2^{-1}$，$b_i = (a_i - b_{i-1}) 2^{-1} (i > 0)$。

求和模 2^t 意义下的结果

可以发现 $b_i = \sum_{j \geq i+2} (-2)^{j-(i+2)} a_j$，当 $j-(i+2) \geq t$ 时 $(-2)^{j-(i+2)} a_j = 0 \mod 2^t$，所以只用计算 $j < t + i + 2$ 即可。

祝大家 WC 取得好成绩!