solution

WerKeyTom_FTD

June 13, 2018

前言

我太弱了

选出n个0到$R-1$间的数，要求两两不同，并使得异或和为0，求方案数。

相互再归的鹅妈妈

要求统计无序方案，由于合法方案中数两两不同，因此我们可以转化为统计有序方案，最后除掉$n!$。

要求统计无序方案，由于合法方案中数两两不同，因此我们可以转化为统计有序方案，最后除掉$n!$。
我们先来解决不要求两两不同要怎么做。

相互再归的鹅妈妈

每一个数都会在某个二进制位获得解放，即在该位之前这个数与R相同，R这一位为1，而这个数在这一位是0。
可以发现，当出现一个数获得解放后，我们不管其后面怎么填，其余数我们乱填，那么这个解放的数由于后面可以乱填一定存在唯一一种方案使得异或和为0。
我们枚举这个位是什么，然后可以简单dp或计数。

怎么考虑加上互不相同呢?
考虑利用这个式子来设计容斥。
$[m = 1] = \sum \prod_{i=1}^{k}(a[i] - 1)!(-1)^{a[i]-1}$
枚举一个m的集合划分，其中k就是这种集合划分方案划分的集合数，而a表示各个集合的大小，它是无序的。

相互再归的鹅妈妈

那么我们可以设计这样的容斥算法：
枚举一个集合划分，同一集合要求相同，不同集合要求不同。
容斥系数即为$\prod_{i=1}^{k}(a[i]-1)!(-1)^{a[i]-1}$。
可以发现对于一个方案，其被该算法计算的次数为$\prod_{i=1}^{k}[a[i]=1]$。
于是我们就解决了本题。

要证明我们利用的容斥式子也很简单。
可以发现它相当于是枚举一种圆排列划分（集合划分，然后每个集合再圆排列），因此其等价于
$\sum_{i=1}^{m} s(m,i)(-1)^{m-i}$
我们知道$x^{\underline{n}} = \sum_{i=0}^{n} s(n,i)(-1)^{n-i}x^i$，令$x = 1$。
然后就能惊讶的发现上述式子等于$1^{\underline{m}} - s(m,0)$，也就是$[m = 1]$。

统计区间能表示成ab^c的数个数，要求b和c大于1，且a小于b。

首先可以证明只需要考虑$c = 2$或$c = 3$的情况。
如果$c = 2k$，且$k > 1$，那么$ab^{2k} = a(b^2)^k$。
如果$c = 2k + 1$，且$k > 1$，那么$ab^{2k+1} = (ab)(b^2)^k$。
显然合法的数一定有$c = 2$或$c = 3$的表示方法。$c = 2$比较简单，
我们下面希望计算只能被表示成$c = 3$的数的个数。

$n=by^3=(by)y^2$，我们找到一个极大的k满足$k^2|by$，那么$n=(\frac{by}{k^2})(yk)^2$。
为了让n无法被表示成$c=2$的情况，必须要有$\frac{by}{k^2}\geq yk$即$k\leq b^{\frac{1}{3}}$。
我们令$c(x)=[xkngf],b(x)=[xk\acute{}gf]$。
答案是$\sum_{b^4\leq n}\sum_{k^3\leq b}\sum_{y=b+1}^{(\frac{n}{b})^{\frac{1}{3}}}c(b)[k^2|by]d(\frac{by}{k^2})$
即$\sum_{b^4\leq n}c(b)\sum_{k^3\leq b}\sum_{y=b+1}^{(\frac{n}{b})^{\frac{1}{3}}}[k^2|by]d(\frac{by}{k^2})$

我们令$g=(k^2,b),k'=\frac{k^2}{g},b'=\frac{b}{g},y'=\frac{y}{k'}$。

式子化为$\sum_{b^4\le n}c(b)\sum_{k^3\le b}\sum_{y'=\frac{b+1}{k'}}^{\frac{(\frac{n}{b})^{\frac{1}{3}}}{k'}}d(b'y')$

我们知道$d(ab)=d(a)d(b)[(a,b)=1]$，将这个代入，然后把$[(a,b)=1]$这样的东西莫比乌斯反演得到最终式子

$\sum_{b^4\le n}c(b)\sum_{k^3\le b}d(b')\sum_{e|b'}\mu(e)\sum_{y'=\frac{b+1}{k'}}^{\frac{(\frac{n}{b})^{\frac{1}{3}}}{k'}}[e|y']d(y')$

盟誓的文艺复兴

$\sum_{b^4\le n} c(b) \sum_{k^3\le b} d(b') \sum_{e|b'} \mu(e) \sum_{y'=\frac{b+1}{k'}}^{\frac{(\frac{n}{b})^{\frac{1}{3}}}{k'}} [e|y']d(y')$

我们可以对于每个$n^{\frac{1}{4}}$内的e，预处理其不超过$n^{\frac{1}{3}}$倍数的前缀和，根据积分计算，级别大约为$n^{\frac{1}{3}}\ log\ n$。

那么计算该式子只需要暴力枚举b,k,e即可，根据积分计算，枚举b,k的复杂度为$O(n^{\frac{1}{3}})$，由于b'没有平方因子，所以可以认为e有$2^{w(n^{\frac{1}{4}})}$。

实际上并不会这么满。于是本题解决。

二维平面随机游走，问经过位置数的方差期望。

要求的是方差，实际就是求所有情况下经过不同位置的个数和以及经过不同位置的平方和，设为d和$d2$。
一共有四种方向向量，任意走法对应一种向量序列。
先预处理$W(i,x,y)$表示长度为i，和为(x,y)的向量序列有多少种。

如何求d，当然是很简单的。
每个坐标显然可以独立统计贡献，我们希望在每个坐标第一次被走到时统计到它。
我们设$g[i]$表示长度为i，且不存在$1 \le j \le i$满足前j个向量前缀和为$(0,0)$。
以后这种类似的情况简单称为"不存在前缀$(0,0)$"。
显然可以容斥：
$g[i] = (w1 + w2 + w3 + w4)^i - \sum_{j=1}^{i} W(j,0,0) * g[i-j]$
那么就有
$d = \sum_{i=0}^{n} \sum_{(x,y)} W(i,x,y) * g[n-i]$
在时刻i我们走到了(x,y)，之后再也没有再走到过。

无限远点的牵牛星

为了简单计算，我们更改$d2$的含义，对于一种情况，对于两个不同的位置（有序）如果都到达过，给$d2$贡献1。那么原本的$d2$等于新的$d2$乘2加上d。
类似的我们设出$R(i,x,y)$表示长度为i，和为(x,y)，不存在前缀$(0,0)$。
R的计算也可以容斥：
$R(i,x,y) = W(i,x,y) - \sum_{j=1}^{i} W(j,0,0) * R(i-j,x,y)$
然后我们设$S(i,x,y)$表示长度为i，和为$(0,0)$，且存在前缀(x,y)。
那么显然
$S(i,x,y) = \sum_{j=1}^{i} W(j,x,y) * R(i-j,-x,-y)$

有了这几个作辅助，我们可以设$F(i,x,y)$表示长度为i的所有情况，前j个向量的和(a,b)为第一次到达，然后接下来$i-j$个向量的和为(x,y)，且在位置i时，$(a+x,b+y)$第一次到达，这样的两个不同位置总数(因此xy不为0)。
可能有点绕。
考虑怎么得到它呢？

无限远点的牵牛星

$F(i,x,y)+=\sum_{j=1}^{i} g[j]*W(i-j,x,y)$
这个是(a,b)是第一次到达的总方案数。
考虑不合法的情况，即(a,b)到达$(a+x,b+y)$后，这个$(a+x,b+y)$已经不是第一次到达了，它可能在第一次到达(a,b)前第一次到达，也可能在第一次到达(a,b)后第一次到达。

$F(i,x,y)- = \sum_{j=1}^{i} g[j] * S(i-j,-x,-y)$
如果在第一次到达(a,b)之前就到达了$(a+x,b+y)$，考虑去掉这种方案，那么相当于第一次到达$(a+x,b+y)$后又走了回来，而且这途中要经过(a,b)。这样一定包含了所有第一次到达(a,b)之前就到达了$(a+x,b+y)$的方案，但是并不止。我们中途去遍历了(a,b)，但是并不一定中途遍历时这个(a,b)才第一次经过。因此我们还会减掉第一次到达(a,b)后，再第一次到达$(a+x,b+y)$，最后在时刻i重新回到$(a+x,b+y)$，这中途还重新回过(a,b)。

无限远点的牵牛星

$F(i,x,y)- = \sum_{j=1}^{i} F(j,x,y) * (W(i-j,0,0) - S(i-j,-x,-y))$
我们考虑第一次到达$(a+x,b+y)$是在第一次到达(a,b)后的不合法方案，注意到上一条式子里我们减掉了第一次到达$(a+x,b+y)$到时刻i间重新回过(a,b)的方案，这里为了不减重要减掉这种方案（即$S(i-j,-x,-y)$）。

那么在得到了F之后，我们很简单就能统计$d2$。
$d2 = \sum_{i=0}^{n} \sum_{(a,b)} \sum_{(x,y)} F(i,a,b) * W(n-i,x,y)$
至此，本题解决。时间复杂度$O(n^4)$。