bitset与二进制

Part 1 bitset优化解异或方程

例1：SPOJ JZPLIT

题意：给出一个$n \times m$的01矩阵，每一次可以对某一个元素进行操作，使得与这个元素在同一行、同一列的元素全部取反。问是否存在一种方案，使得所有的元素最终变成0，并且输出任意一种方案中，每个元素的操作次数对2取模的结果。

Solution：用$a_{i,j}$表示初始矩阵中的元素。首先设$x_{i,j} = 0/1$表示第i行第j列是否进行操作。

设$sumx_i$为$x_{i,1} \bigoplus x_{i,2} \bigoplus x_{i,2} \cdots \bigoplus x_{i,m}$，$sumy_i = x_{1,i} \bigoplus x_{2,i} \bigoplus \cdots \bigoplus x_{n,i}$。

考虑某两个格子(i_1, j_1)和(i_2, j_2)，我们可以得到方程：

$$\begin{cases} a_{i_1,j_1} \bigoplus x_{i_1,j_1} \bigoplus sumx_{i_1} \bigoplus sumy_{j_1} = 0 \\ a_{i_1,j_2} \bigoplus x_{i_1,j_2} \bigoplus sumx_{i_1} \bigoplus sumy_{j_2} = 0 \\ a_{i_2,j_1} \bigoplus x_{i_2,j_1} \bigoplus sumx_{i_2} \bigoplus sumy_{j_1} = 0 \\ a_{i_2,j_2} \bigoplus x_{i_2,j_2} \bigoplus sumx_{i_2} \bigoplus sumy_{j_2} = 0 \end{cases}$$

把这四个方程全部异或起来得到：

$$a_{i_1,j_1} \bigoplus a_{i_1,j_2} \bigoplus a_{i_2,j_1} \bigoplus a_{i_2,j_2} \bigoplus x_{i_1,j_1} \bigoplus x_{i_1,j_2} \bigoplus x_{i_2,j_1} \bigoplus x_{i_2,j_2} = 0$$

这意味着我们可以用$x_{i_1,j_1}, x_{i_1,j_2}, x_{i_2,j_1}$把$x_{i_2,j_2}$表示出来。

那么，我们只需要把$x_{1,i}$和$x_{i,1}$，也就是第一行和第一列的所有元素都设出来，方程中需要用到的、不在第一行第一列的变量，都可以用第一行、第一列的变量表示。解出方程后，矩阵中其他位置的元素自然也得到了。

例2：SPOJ JZPLIT2

题意：给出一个$n \times m$的矩阵，其中有一些位置是障碍，其他位置是0/1。对一个位置进行操作后，这个位置上的数取反，并且与这个位置在同一行、同一列，并且之间没有障碍的格子取反。问是否存在一种合法的方案，使得最终所有不是障碍的位置上的数都变成0，并输出方案。$n, m \leq 300$，障碍的数量$\leq max(n, m)$。

Solution：通过与上一道题相似的方法，我们如果把第一行和第一列的所有格子，以及每一个障碍的右边，下边，右下边的三个格子全部设出来，我们就可以用这些未知数表示出整个矩阵中任意一个位置的元素。

具体地，假如我们已经用设的变量表示出了$x_{i-1,j-1}, x_{i,j-1}, x_{i-1,j}$，那么通过上一道题的结论，我们有

$$x_{i-1,j-1} \bigoplus x_{i-1,j} \bigoplus x_{i,j-1} \bigoplus x_{i,j} \bigoplus a_{i-1,j-1} \bigoplus a_{i-1,j} \bigoplus a_{i,j-1} \bigoplus a_{i,j} = 0$$

然后就可以列出方程求解了。

Part2 bitset优化只有0/1取值的背包

例1：bzoj3687 简单题

给出n个元素$a_1, a_2, \cdots a_n$，问这n个元素的所有子集的算术和的异或和。$n \leq 1000, \sum a_i \leq 2000000$

Solution：考虑计算每一种个x，计算有多少个子集的算术和等于它。又由于是异或，所以我们只需要关心这样的子集数量的奇偶性。设$dp[i][j]$为前i个元素，算术和为j的子集数量的奇偶性。可以用bitset优化转移。

例2：AGC020 C Median Sum

给出n个元素$a_1, a_2, a_3 \cdots a_n$。考虑它的$2^n - 1$个非空子序列，每个子序列中元素的和$s_1, s_2 \cdots s_{2^n-1}$，求$s$的中位数。$n, a_i \leq 2000$

Solution：设$sum = \sum a_i$，那么对于任意一个子集和x，存在$sum - x$与之对应，但是由于这里没有计算空集，因此，答案是大于$\frac{sum}{2}$的最小的子集和，可以转化成求它的补集，小于$\frac{sum}{2}$的最大的子集和。用bitset优化背包就可以了。

Part3 二进制相关

例1：CF878D

每一个生物有n种属性，每一种生物的每一种属性都有一个属性值。初始的时候有k种生物。有三种操作：1.由第x种生物和第y中生物创造出一种新的生物，这种生物的每种属性的属性值为x和y在这种属性上的属性值的max。2.由第x种生物和第y中生物创造出一种新的生物，这种生物的每种属性的属性值为x和y在这种属性上的属性值的min。3.查询第x种生物的第y中属性的属性值。$n, m \leq 10^5, k \leq 12$

Solution：如果属性值只有0/1的话将会特别容易做：1就是每一种属性进行或运算，2就是与运算。如果二分一下，好像就可以转化成0/1了。但是n特别大，如果每新加入一个生物的时候把它的属性值都处理出来肯定不行。

二分完答案过后，最初的k个生物的属性值变成了0/1，我们可以由最初的k个生物的这个0/1的属性推出后面的每一个生物的属性值是0还是1。考虑到这种初始状态只有2^k个，我们可以处理对于每一种初始状态，每一个生物会是0/1。最后对于枚举查询的属性值枚举一下会和最初的k个生物中的哪一个相等就可以了。

例2：AGC006D Median Pyramid Hard

给一个长度为n（保证n为奇数）的序列a，每一次操作把a_i变成一个长度为$n-2$的序列b，其中$b_i = median(a_{i-1}, a_i, ai+1)$，其中median表示中位数。问操作到$a$中只有一个元素的时候，这个元素是多少。$n \leq 2 \times 10^5$

Solution：如果序列中只有0/1的话，就会简单很多。可以通过二分答案来达到这个效果

每一次取中位数，相当于取三个数中出现次数最多的元素。

通过观察可以发现，如果某一段是01相间，而两边都是连续的两个0或者1，那么每经过一次操作，01相间的这一段的长度就会减少1。而任意的连续的单独由0或者1构成的、长度大于1的一段，每经过一次操作，长度就会增加2。

总结起来，最终剩下的那个元素，要么是离$\lceil \frac{n}{2} \rceil$最近的一段连续的、长度大于等于2的0/1，要么就是a_1——此时整个序列中不存在长度大于等于2的、连续的0或者1。

Burnside引理相关

> - 如果所有长度相同的循环节都是一样的，即我们在计算不动点个数的时候不关心循环节中具体有那些元素，而只关心每一种长度的循环节各有多少个，那么我们可以考虑枚举所有的划分。

> - 当n取到49的时候，n个元素的划分数为173525，枚举划分的复杂度是可以接受的。
> - 对于排列，假设我们枚举的划分的这种方案中，l_i这种长度的循环节出现了k_i次，那么符合这种划分方案的排列的数量是$\frac{n!}{\prod k_i! l_i^{k_i}}$

例1：bzoj1815 有色图

给一个完全的无向图，你需要给图中的每一条边涂色。两张图定义为本质相同的，当且仅当其中一张图通过对顶点重新编号可以变得和另一张图完全一样。问本质不同的图的数量。$n \leq 53$

Solution：根据前面的分析，我们可以枚举置换中，每个长度的循环节的数量。

图中的边被分成了两类：某一个循环节内部的顶点之间的边，和两个属于不同循环节的顶点之间的边。

考虑同一循环节内的边，如果原来的边是(u,v)，那么一轮置换后就会变成(p_u,p_v)，显然(u,v)和(p_u,p_v)属于同一个循环节，染的颜色应该相同。我们可以枚举u,v在循环节上的“间隙”，即u经过几次这个置换会变到v上。“间隙”相同的边应该然成相同的颜色，而对于某一个间隙，这样的边的数量恰好为len，即循环节长度。由此可以推出，这个循环节内部，边染色的循环节数量为$\lfloor \frac{len}{2} \rfloor$。（除以2是因为图为无向图）

循环节之间的边，可以推出边的循环节数是$gcd(len_1,len_2)$。

变式：本质不同的有向图计数

我们考虑枚举所有的有向边，给它们染色，颜色1表示出现在图中，颜色2表示不出现在图中。仍然枚举循环节划分。一个循环节内的边的贡献是$len-1$，即边会构成$len-1$个循环节；然后枚举**从一个循环节到另一个循环节**的边，注意这是有方向的，贡献是$gcd(len_1,len_2)$。

例2：codechef ADIMAT

问有多少个n行m列的本质不同的01矩阵。其中，两个矩阵是本质相同的，当且仅当通过交换其中一个矩阵的若干行，或者若干列，可以使得两个矩阵变得一样。答案对10^9+7取模。$nm \leq 550$

Solution：考虑一种暴力的做法：暴力枚举行的置换的循环节划分，枚举列的置换的循环节划分。此时我们会发现整个矩形被分成了 行的循环节数 $\times$ 列的循环节数 个小的矩形，每个小的矩形内部的格子循环。观察可得，每一个这样的小的矩形，内部格子的循环节数量恰好为长和宽的最大公约数。

进一步观察发现，对于列的置换，每一种长度的循环节的贡献（无论是对于计算符符合要求的排列数量，还是对于计算矩阵格子的循环节数量），与列的置换中其他长度的循环节是无关的。对于行也是如此。注意较小的一维不会超过23，于是我们可以暴力枚举较小的一维的划分，然后用dp解决另一维。

杂题

codechef SFXPAL

给出字符集大小S，字符串长度N，以及模数M，问有多少个长度为n的字符串，满足没有任意一个后缀是回文串。

Solution：递推式$f_i=S\cdot f_{i-1}-f_{\lceil \frac{i}{2} \rceil}$。

证明递推式的正确性，只需要证明任意一个不含有后缀回文的串X，我们将它翻转后拼接，可以得到一个除了整个完整的串之外，不存在后缀回文的字符串X^TX。

如果有的话，这个后缀回文的长度一定大于字符串长度的一半。设为$\cdots(Y|\cdots Y^T)$，其中|表示翻转对称轴，括号括起来的表示回文后缀。

根据翻转和回文的定义，我们可以推出$\cdots(Y|Y^T\cdots YY^T)$，而YY^T一定是回文，这与我们X不含有后缀回文的条件矛盾。

sgu537 Divisibility

给一个字符串，字符串中的每一个字母可以对应0到9之间的数字，不同的字符对应的数字必须不同。问所有对应的方式得到的所有数字的gcd是多少。保证字符串中的字符的数量不超过10。

Solution：以abbaca为例，我们构造：$a_a=100101, a_b=11000, a_c=10$，那么任意一个可以得到的数字，一定可以被表示成$k_1a_1+k_2a_2\cdots$的形式。

我们考虑让它们相减。如果我们让两个数字$k_1a_1+k_2a_2+k_3a_3$和$k_1a_1+k_2a_2+(k_3+1)a_3$相减，我们就可以得到a_3。进一步归纳发现，如果字符串中字符的种数小于等于9，那么答案就是$gcd(a_1,a_2,a_3\cdots)$。

而对于字符的种数等于10的情况，我们考虑构造$a_1+2a_2+3a_3$与$2a_1+a_2+3a_3$，两个做差可以得到a_1-a_2，推广下来可以得到答案为$a_1,a_2\cdots$两两的差的gcd。

介绍了另一种gcd的写法：1）如果x,y都是偶数，那么返回$gcd(\frac{x}{2},\frac{y}{2})\cdot 2$。2）如果其中一个是偶数（假设是$x$），那么返回$gcd(\frac{x}{2},y)$。3）否则，返回$gcd(x-y,y)$。此时$x-y$一定是偶数。

codechef ADITREE

给一棵树，树上的边的边权都是1，树上的节点与开灯和关灯两个状态。每一次会翻转两个节点的状态，而你需要输出，当前状态下，将开灯的房间（保证有偶数个）两两配对，每一对的距离的和的最小值。$n\le 2.5\times 10^5$

Solution：如果一条边被经过的次数大于2，那么我们一定可以把两条经过它的路径异或一下得到两条更短的路径。因此，一条边的会产生1的贡献，仅当它两边的开灯的节点的数量都是奇数。修改相当于翻转一条路径上的边。可以直接树剖维护。

codechef EBAIT

有n个车队，第i个车队包含V_i辆车。如果i为奇数，则会对票数产生$V_i\times X$的贡献，否则会产生$-V_i\times Y$的贡献。现在要求在第n个车队开来之前，票数始终为正数（严格大于0），并且在第n个车队开来过后，票数变成负数。你需要确定X,Y的值，并满足$X\le C_1, Y\le C_2$，问是否存在一种方案，并输出方案。

Solution：把每一个车队转化成$aX+by>0$的限制，问题最终就转化成了，在两个数之间，是否存在一个分数，它的分子和分母都不大于C_1,C_2。

考虑到把分子变大并不会让分母变小，所以我们只需要让分母尽量小就可以了。

欧几里得算法的应用 P11

USACO 2019 Feb PT C Mowing Mischief (luogu P5244)

一个$T\times T$的网格（其中有$(T+1)\times(T+1)$个格点），其中有n个格点上有花，保证每一行、每一列至多有一朵花。你需要选择一些花，满足这些花按照x排序后，y也是严格递增的。你还必须选择$(0,0)$和$(T+1,T+1)$。你的选择方案必须保证，选择的花的数量是最大的，在满足这个条件的基础上，假如选择的花的坐标按照x排序后的坐标是$(x_0,y_0),(x_1,y_1)\cdots(x_k,y_k)$，你需要最大化$\sum(x_i-x_{i-1})(y_i-y_{i-1})$。$T\le 10^6, n\le 2\times 10^5$

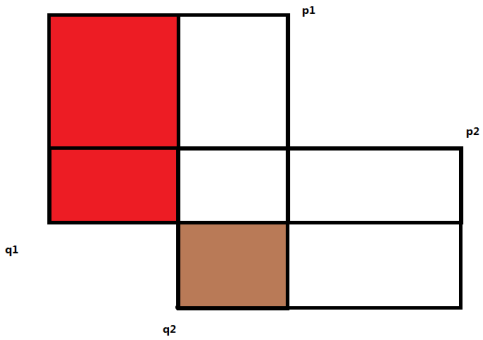
Solution：首先，我们按照以这个点结尾的最长上升子序列的长度，把所有的n朵花分层。同一层中的点显然只能够转移到下一层的点。$O(n^2)$的暴力是显然的。

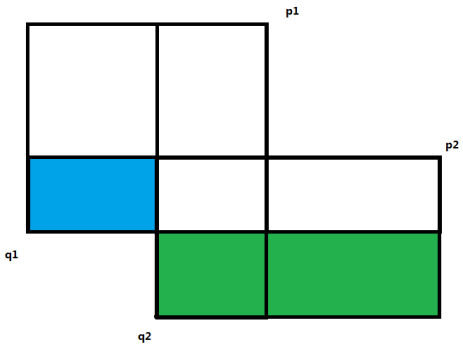
结论1：对于同一层中的点，如果把它们按照x升序排序，那么它们的y是降序的。

结论2：如果A是某一层中的一个节点集合，B是下一层的一个节点集合，并且满足，从A的任意一个点都可以转移到B的任意一个点。那么最优的决策点满足，对于B中的横坐标依次递增的点，它们在A这边的决策点的横坐标一定是依次下降的。

结论2的证明：设p_1, p_2是下一层的两个点且满足p_1在p_2的左上方，它们的最优决策点分别为q_1, q_2且满足q_1在q_2的左上方。用$S(a, b)$表示$(a_x - b_x)(a_y - b_y)$。那么就有：

$$
\begin{aligned}
f_{q_1} + S(p_1, q_1) &< f_{q_2} + S(p_1, q_2) \\
f_{q_2} + S(p_2, q_2) &\le f_{q_1} + S(p_2, q_1) \\
\Rightarrow S(p_1, q_1) + S(p_2, q_2) &\le S(p_1, q_2) + S(p_2, q_1) \\
\text{即 } S(p_1, q_1) - S(p_1, q_2) &\le S(p_2, q_1) - S(p_2, q_2)
\end{aligned}
$$





从上图中明显可以看出，红色-褐色 > 蓝色-绿色，上式是不可能成立的。由此结论得证。

但是任意两层之间的点，可能并不满足**下一层的点可以到达上一层的任意一个点**。考虑对上一层的点建一棵线段树，把下一层的点作为标记打在它覆盖的第一层的点上，然后对线段树上的所有区间利用决策单调性求解就可以了。

luogu P5155 Balance Beam

有一根平衡木，上面有$0, 1, 2 \cdots n+1$这些点，一旦走到0或者$n+1$就会立刻掉下去，收益为0。当你走到第i个点的时候，你可以选择从这个点离开，获得a_i的收益；你也可以$\frac{1}{2}$的概率向左走一步，以$\frac{1}{2}$的概率向右走一步。问最优策略下，从每一个点出发能够得到的最大收益。最优策略的含义是，选择期望收益最大的策略。$n \le 10^5$

Solution：考虑一个问题，从i出发，向左和向右走一步的概率各是$\frac{1}{2}$，到达0或者n就停止。问最后停在n的概率。

$p_i = \frac{(p_{i-1}+p_{i+1})}{2}$，因此这是一个等差数列，因此从$i$走到$n$的概率是$\frac{i}{n}$。并且，如果概率不是$\frac{1}{2}$，则$p_i$差分数组是一个等比数列。

对于i，若存在$r \ge i, l \le i$，且我们的策略是走到a_l或者a_r立刻停止，那么收益期望就是$\frac{a_l(i-l)+a_r(r-i)}{r-l}$。这个式子相当于一条经过$(l, a_l)$和$(r, a_r)$的直线，在$x = i$的时$y$的取值。又由于我们取收益期望最大，所以本质上，我们需要对(i, a_i)维护一个凸壳，第k个点的答案就是$x = k$与凸壳的交。

luogu P5156 Sort It Out

有一个长度为n的排列。Farmer John将会选择一个集合，然后不断地对这个集合的元素进行调整：具体地，如果他要调整第i个元素，它将会把i与左右的元素交换直到i大于了它左边的第一个元素，并且小于了它右边的第一个元素。他会不断对集合中的元素进行调整直到集合中的元素都无法再调整。问所有可以使排列最终有序的集合中，大小最小是多少，以及满足大小最小的字典序第k小的集合是哪一个。$n \leq 10^5, K \leq 10^{18}$

Solution：不在集合中的元素，相对顺序不会改变。而在集合中的元素，与左右元素的顺序是对的。因此，只要我们选出的集合的补集是一个上升子序列就可以了。

选出集合第k小可以转化成补集的第k大。可以把每个元素选或不选转化成一个二进制数，那么比较两个集合的时候是从高位到低位比较，与两个数的大小的比较是相同的。

求出从每个点开始的**最长**上升子序列个数，然后从第一位开始一位一位地确定就可以了。确定每一位的复杂度是与这一层中点的数量有关的，由于每个点只会属于一层，所以这个逐位枚举的过程的复杂度是$O(n)$的。

luogu P5123 Cowpatibility

有n头奶牛，每一头奶牛有5种喜欢的冰淇淋口味，每一种冰淇淋口味用$[1, 10^6]$之间的一个正整数表示。问有多少对奶牛，它们没有共同的喜欢的冰淇淋口味。$n \leq 5 \times 10^4$

Solution：考虑容斥，枚举两个奶牛共同喜欢的冰淇淋集合。这里由于每个奶牛只会对2^5个集合产生贡献，所以复杂度是$O(n2^5)$的。可以用hash把冰淇淋集合映射为一个整数。

luogu P5204 Train Tracking 2

有一个长度为n（$n \leq 10^5$）的序列，序列中的元素都是$[1, 10^9]$中的数。显然它有$n-k+1$个长度为k的区间。给出这$n-k+1$个区间的区间最小值$c_i, c_2 \cdots c_{n-k+1}$，问有多少个序列满足条件。答案对$10^9+7$取模。

Solution：观察可以发现，如果$c_i \neq c_{i+1}$，那么我们一定就可以据此确定原序列中的一个位置：

- 如果$c_i > c_{i+1}$，那么新的这个最小值一定是由a_{i+k}贡献的，即我们可以确定$a_{i+k} = c_{i+1}$。
- 如果$c_i < c_{i+1}$，那么新的这个最小值一定是由a_i贡献的，我们可以确定$a_i = c_i$。

而这些确定位置的之间的区间，我们对它们的限制是这样的：所有的位置上的数都不能够小于x，并且每k个连续的数中，x必须出现至少1次。可以设$dp[i]$为第i个位置填了x的方案数，则我们可以枚举上一个x出现的位置，然后可以再加一个前缀和优化。总时间复杂度$O(n)$。