杂题选讲

CF1592F

对于一个二维01矩阵，有两种操作方式：

翻转包含 (1,1) 的矩阵，代价为1

翻转包含 (n,m) 的矩阵，代价为2

求把矩阵全修改为 0 的最小代价

$n,m \leqslant 500$

令原矩阵为 c ，设矩阵 a ，且 $a_{i,j}=(c_{i,j}+c_{i+1,j}+c_{i,j+1}+c_{i+1,j+1}) \mod 2$

也可换一种表述

$a_{i,j}=c_{i,j}\ xor\ c_{i+1,j}\ xor\ c_{i,j+1}\ xor\ c_{i+1,j+1}$

至此，**将 c 全修改为 0 等价于将 a 全修改为 0**

对于第一种操作 (i,j) ，等价于翻转 $a_{i,j}$

对于第二种操作 (i,j) ，等价于翻转 $a_{i-1,j-1}$， $a_{i-1,m}$ ， $a_{n,j-1}$， $a_{n,m}$

至此，此题可做

[NWRRC2017] Dividing Marbles

给出四个正整数 d_1, d_2, d_3, d_4,表示一个正整数 $n = 2^{d_1} + 2^{d_2} + 2^{d_3} + 2^{d_4}$。

维护一个集合 S，初始 S 中初始只有数 n。

每次可以进行如下操作:

选择 S 中任意一个大于 1 的数，把它拆成两个正整数插入集合，然后从集合中把这个数删去。

当 S 中只剩下 1 时，不再进行操作。

询问最少的操作次数。

共有 T 组数据。

$T \leqslant 500, d_1, d_2, d_3, d_4 \leqslant 20$

Hint

考虑一下答案的上界?

```
输入
2
1 0 1 0
0 1 2 3
```

```
输出
3
6 2 4
4 2 2
2 1 1
5
15 10 5
10 5 5
5 1 4
4 2 2
2 1 1
```

sol

考虑这个拆分的过程，不难发现这个东西其实是寻找长度最短的计算出 n 的 加法链。

不严谨通俗地描述一下这个东西：

设一段长度为 k 的序列 a，其中 $a_1 = 1, a_k = n$，同时对于 2 到 k 的每一个数 i，都能在前 $i-1$ 个数里选出两个（可以重复）的数，使得它们的和等于 a_i。

那么这就是一条长度为 k 的计算出 n 的加法链。

设 n 的二进制表示位数为 s，二进制表示中为 1 的位数为 t，显然 $t \le 4$。

对于这题我们显然可以找出一种构造方案，满足它的加法链长度最多为 $s+t-1$，介绍如下：

- 把 n 拆成 $lowbit(n)$ 和 $n - lowbit(n)$，直到 $n = lowbit(n)$ 为止，需要操作 $t-1$ 次。
- 不断让 n 除以 2 除到 1 为止，需要 $s-1$ 次。

步骤二中会顺便将所有的 $lowbit$ 去重，因此一定只剩下一个 1，所以这个做法是正确的。

最后再把 1 加上，所以加法链的长度最长是 $s+t-1$, 但是也有特殊例子，比如 $15 = (1111)_2$，但是它的最短加法链长度为 6。$(\{1,2,4,5,10,15\})$

考虑这个东西的性质：

1. $a_i \geq \lfloor \frac{a_{i+1}}{2} \rfloor (1 < i < k)$，这个比较显然。
2. $a_i \geq 2^{i-t+1}$，这个东西有点不显然，下面证明这个东西：

   我们知道本题当中 $k \leq s+t-2, a_k = n \geq 2^{s-1}$。

   稍微做一下变形：$k-t+1 \leq s$，两边同时作为指数，就可以得出 $a_k \geq 2^{k-t+1}$，再结合性质一，可以得到结论二。

考虑 $t \leq 4$，因此可以得到 $a_i \geq 2^{i-3}$，考虑直接爆搜，发现不够。

考虑到取等号的时候我们可以直接构造方案，因此我们只需要爆搜满足 $a_i \geq 2^{i-2}$ 的所有情况就行，这时爆搜只需要 0.2s 左右，很快。

P8340 山河重整

给出 n，求 {1,2,3,...,n} 有多少子集满足，对于 1~n 的每一个数都可以被拼凑出来（每一个数最多选一次），答案对 m 取模。

$n \leq 5 \times 10^5, 2 \leq m \leq 1.01 \times 10^9$

Hint1：得到一个集合合法的充要条件。

Hint2：考虑容斥。

Hint3：通过一些观察降低复杂度。

首先，对于一个集合，假如它可以拼凑 $[1,k]$ 的所有数，那么加入 x 后会变成 $[1,k+x]$，前提是 $x\le k$。

所以有一个 $O(n^2)$ 的 dp 做法，设 $f_{i,j}$ 表示从小到大考虑前 i 个元素后，可以拼凑出 $[1,j]$ 所有数的方案数，转移是容易的。

但是难以优化，所以我们可以尝试容斥，具体来说，枚举第一个不能被拼凑的位置，用总方案数去减去这些方案。

设 f_i 表示使用前 i 个数，可以拼凑出 $1\sim i$ 所有数，且这些数总和为 i 的方案数，那么对答案的贡献是 $f_i 2^{n-i-1}$，问题变为快速计算 f。

我们发现 f_i 为 所有 i 的**互异拆分数**减去**存在** $[1,i]$ **的数拼不出来的方案**，后者可以利用之前计算出来的 f 计算，前者也可以 dp，不过这两个部分都是 $O(n^2)$ 的，所以先从相对简单的第一部分开始优化。

注意到一个数 i 的互异拆分数最多选出 $O(\sqrt{i})$ 个数，所以不妨枚举当前选了多少个数字，做完全背包，这一部分复杂度降为 $O(n\sqrt{n})$。

f_j 对 f_i 的贡献也形如 $[j+1,i]$ 内的数做 01 背包，并且发现，只有 $2j \le i$，f_j 才对 f_i 有贡献，所以可以分治计算 f。具体来说，对于分治区间 $[l,mid],[mid+1,r]$，只有前半部分对后半才有贡献，而这一部分贡献可以类似之前的方式优化。

总复杂度为 $O(n\sqrt{n}+\frac{n}{2}\sqrt{\frac{n}{2}})+\ldots=O(n\sqrt{n})$。

Utakotoba

对于一个长度为 n 的序列 A， 你可以进行如下操作：

选择两个位置 x,y ， 满足 |x-y|=1 ， 将 A_x 赋值为 A_x xor A_y。

现在你想用不超过 150n 次操作，将序列 A 变为给出的序列 B 。

数据保证有解。

$n \le 20000$

发现你的操作可以逆回去。

然后这个东西看起来就比较像线性基。

所以我们的思路就是将 A 和 B 化为同一个线性基
，先正序进行 A 的操作，再逆序进行 B 的操作。

线性基要求一模一样的话可以考虑高斯消元。

我们可以用三次操作实现两个数的交换。

然后这题就做完了。

精细思考后挺好实现的。

[AGC023F] 01 on Tree

给出一棵 n 个节点的树，以及一个空序列。

每个节点上有一个取值在 {0,1} 中的数。

每次你可以选择没有父亲节点的点删除，并且将这个节点上的数字放在当前数列末尾。

请你求出这个数列可能得到的最小逆序对数。

$$n \le 2 \times 10^5$$

题目让我们从上往下删除来得到顺序，考虑倒过来改成合并，可以发现这没有本质区别。

我们唯一不能确定的地方就是一个点有多个子树的情况，我们想要确定一个顺序使得合并上来新产生的逆序对最小。并且可以发现这里面的顺序不会影响外面其他的贡献，因此启发我们用 Exchange Argument 来做。

对于两个连通块 A, B，令 A_i 表示 A 中 i 的数量，B 同理。

A 在 B 前面，$A_1 \times B_0$。

B 在 A 前面，$B_1 \times A_0$。

当 $A_1 \times B_0 < B_1 \times A_0$ 即 $\frac{A_1}{A_0} < \frac{B_1}{B_0}$ 时，我们会优先合并 A，否则先合并 B。

因此初始时我们让每个点是一个连通块，每次取出最小的这个值来和父亲合并，用可删堆或者 `set` 维护就可以了。

注意分母为 0 的情况。

时间复杂度 $O(n \log n)$。

HolyK's Land

给定一棵 n 个点的树，和 m 条链。

q 组询问，每组询问给定 l,r，求与到编号在 l,r 之间的链距离不超过 1 的点数。

允许离线。

$$n, m \le 10^5, q \le 5\ \times 10^5$$

Hint

考虑离线扫描线，每次查询与编号大于 x 的链距离不超过 1 的点数。

考虑一种与本题相适应的编号方式。

当递归到一个点时，先将其所有儿子依次编上号，然后递归重儿子，这样距离一条链不超过 1 的点集在编号上形成 $O(\log)$ 段区间，珂朵莉树维护即可。

复杂度 $O(n\log^2 n)$