杂题选讲

Polygon-yg

ASDFZ

2020.10.31

「CF1270G」Subset with Zero Sum

给你n个整数$a_1, a_2, \ldots, a_n$,其中a_i满足$i - n \leq a_i \leq i - 1$请找到一个这些整数的一个非空子集，使得它们的和为0。可以证明在给出的条件下一定存在这样的子集。如果有多个子集满足条件，输出其中一个。

「CF1270G」Subset with Zero Sum

给你n个整数$a_1, a_2, \ldots, a_n$,其中a_i满足$i - n \le a_i \le i - 1$请找到一个这些整数的一个非空子集，使得它们的和为0。可以证明在给出的条件下一定存在这样的子集。如果有多个子集满足条件，输出其中一个。

$1 \le n \le 10^6$

「CF1270G」Subset with Zero Sum

我们可以发现$1 \le i - a_i \le n$。

「CF1270G」Subset with Zero Sum

我们可以发现$1 \le i - a_i \le n$。

于是我们从i向$i - a_i$连一条有向边，如果我们走过这条边表示将这个数加入子集，然后我们去寻找一个环，显然将环上的数选择是可行的。

「CF1270G」Subset with Zero Sum

我们可以发现$1 \leq i - a_i \leq n$。

于是我们从i向$i - a_i$连一条有向边，如果我们走过这条边表示将这个数加入子集，然后我们去寻找一个环，显然将环上的数选择是可行的。

由于一共n个点，n条边，所以一定会存在环。

「IOI2020」装饼干

有k种饼干，第i种美味值为2^i，有a_i个。现在要选出一些饼干装进x个盒子，要求每个盒子中饼干的美味值之和相同，记为y。问一共有多少种可以组合出的y。

「IOI2020」装饼干

有k种饼干，第i种美味值为2^i，有a_i个。现在要选出一些饼干装进x个盒子，要求每个盒子中饼干的美味值之和相同，记为y。问一共有多少种可以组合出的y。

$1 \le k \le 60, 1 \le x, \sum a_i \le 1e18$

「IOI2020」装饼干

我们按二进制一位一位处理，假设我们用若干饼干凑出了x个盒子的前i位，现在我们要凑第$i+1$位。

「IOI2020」装饼干

我们按二进制一位一位处理，假设我们用若干饼干凑出了x个盒子的前i位，现在我们要凑第$i+1$位。

设除去我们用的饼干后，美味值小于等于2^{i+1}的饼干美味值之和为w，那么我们显然可以在$i+1$位至多装出$\lfloor \frac{w}{2^{i+1}} \rfloor$个盒子。这样就可以判断一个美味值是否可以组合出了。

「IOI2020」装饼干

我们按二进制一位一位处理，假设我们用若干饼干凑出了x个盒子的前i位，现在我们要凑第$i+1$位。

设除去我们用的饼干后，美味值小于等于2^{i+1}的饼干美味值之和为w，那么我们显然可以在$i+1$位至多装出$\lfloor \frac{w}{2^{i+1}} \rfloor$个盒子。这样就可以判断一个美味值是否可以组合出了。

我们将这个过程进行DP即可。

「IOI2020」植物比较

有一个长度为n的排列，把它排成一个环，给定每个数顺时针方向后$k-1$个数中有多少比它大。然后q组询问，每次询问两个位置上的数的大小关系，或者判断出关系无法确定。

「IOI2020」植物比较

有一个长度为n的排列，把它排成一个环，给定每个数顺时针方向后$k-1$个数中有多少比它大。然后q组询问，每次询问两个位置上的数的大小关系，或者判断出关系无法确定。

$n, q, k <= 2e5$

「IOI2020」植物比较

我们考虑寻找任何一个可能排列的过程。

「IOI2020」植物比较

我们考虑寻找任何一个可能排列的过程。

令a_i表示后面顺时针$k-1$个有多少比i大，我们每次可以把a_i为0的挑出来，且逆时针$k-1$个中不存在a_i为0的，显然这样的数才可能是最大值，我们可以把它删掉并把逆时针$k-1$个a_i减一。

「IOI2020」植物比较

我们考虑寻找任何一个可能排列的过程。

令a_i表示后面顺时针$k-1$个有多少比i大，我们每次可以把a_i为0的挑出来，且逆时针$k-1$个中不存在a_i为0的，显然这样的数才可能是最大值，我们可以把它删掉并把逆时针$k-1$个a_i减一。

这样我们可以发现每次删除的数比它后$k-1$个数中未删除的数一定都要大，比删除的数一定小，于是我们可以得出任意相邻k个数的大小关系，建出图跑BFS即可$O(n^3)$，这样，如果两个点之间无路径可达，即无法判断，否则就判断出大小关系了。

「IOI2020」植物比较

我们考虑寻找任何一个可能排列的过程。

令a_i表示后面顺时针$k-1$个有多少比i大，我们每次可以把a_i为0的挑出来，且逆时针$k-1$个中不存在a_i为0的，显然这样的数才可能是最大值，我们可以把它删掉并把逆时针$k-1$个a_i减一。

这样我们可以发现每次删除的数比它后$k-1$个数中未删除的数一定都要大，比删除的数一定小，于是我们可以得出任意相邻k个数的大小关系，建出图跑BFS即可$O(n^3)$，这样，如果两个点之间无路径可达，即无法判断，否则就判断出大小关系了。

我们发现图中边数过多，于是考虑缩小图的规模，显然我们只要在相邻k个数中，对每一个数找到它的前驱后继连边即可，复杂度变为$O(n^2)$。

然后我们来思考一条x到y的路径长什么样，一个结论是存在一条路径使得你一直沿着顺时针方向走或沿着逆时针方向走，因为我们如果是从顺时针方向接近y的，我们逆时针走一步只会让你要判断的a_x更大，反而更劣。

「IOI2020」植物比较

然后我们来思考一条x到y的路径长什么样，一个结论是存在一条路径使得你一直沿着顺时针方向走或沿着逆时针方向走，因为我们如果是从顺时针方向接近y的，我们逆时针走一步只会让你要判断的a_x更大，反而更劣。

另外因为只有前驱和后继有用，于是我们只保留每个点后$k-1$个数中的前驱和前$k-1$个数的前驱即可。

「IOI2020」植物比较

然后我们来思考一条x到y的路径长什么样，一个结论是存在一条路径使得你一直沿着顺时针方向走或沿着逆时针方向走，因为我们如果是从顺时针方向接近y的，我们逆时针走一步只会让你要判断的a_x更大，反而更劣。

另外因为只有前驱和后继有用，于是我们只保留每个点后$k-1$个数中的前驱和前$k-1$个数的前驱即可。

每一次倍增判断是否能从两侧跳到与y距离小于k的点上去，如果能再判一下大小关系即可，复杂度$O(nlogn)$。

「第13回日本情報オリンピック本選」E - 切り取り線

给定 $w \times h$ 的矩形内的 n 条线段，保证每条线段平行于矩形边界，问这些线段将这个矩形分成了多少个联通块。

「第13回日本情報オリンピック本選」E - 切り取り線

给定 $w \times h$ 的矩形内的 n 条线段，保证每条线段平行于矩形边界，问这些线段将这个矩形分成了多少个联通块。

$n \leq 10^5, w, h \leq 10^9$

首先肯定要离散化，这没有问题。

「第13回日本情報オリンピック本選」E - 切り取り線

首先肯定要离散化，这没有问题。

我们可以先考虑一个十分暴力的做法，我们将矩形化成 1×1 的小方格，然后再找联通块。紧接着我们可以发现这之中的不少格子都是可以合并的，最后合并后剩下的个数就是联通块数。

「第13回日本情報オリンピック本選」E - 切り取り線

首先肯定要离散化，这没有问题。

我们可以先考虑一个十分暴力的做法，我们将矩形化成 1×1 的小方格，然后再找联通块。紧接着我们可以发现这之中的不少格子都是可以合并的，最后合并后剩下的个数就是联通块数。

如果暴力全部拆开来的话我们会得到一个非常暴力的 $O(n^2)$ 的做法。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

如果一些线段并没有相交，我们可以将它们分开来计算，这样对于答案是没有影响的。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

如果一些线段并没有相交，我们可以将它们分开来计算，这样对于答案是没有影响的。

至于这个东西怎么处理我们可以用并查集，但是并查集比较难求线段交，所以我们可以用线段树来实现。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

如果一些线段并没有相交，我们可以将它们分开来计算，这样对于答案是没有影响的。

至于这个东西怎么处理我们可以用并查集，但是并查集比较难求线段交，所以我们可以用线段树来实现。

然后欧拉定理可以发现，整张图的联通块个数就可以表示成交点个数加上联通块个数减去线段个数。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

如果一些线段并没有相交，我们可以将它们分开来计算，这样对于答案是没有影响的。

至于这个东西怎么处理我们可以用并查集，但是并查集比较难求线段交，所以我们可以用线段树来实现。

然后欧拉定理可以发现，整张图的联通块个数就可以表示成交点个数加上联通块个数减去线段个数。

接下来唯一麻烦的就是如何求交点，这个可以直接扫描线轻松地解决。

「第13回日本情報オリンピック本選」E - 切り取り線

发现边界非常麻烦，我们把边界拆成四条线段来做。

如果一些线段并没有相交，我们可以将它们分开来计算，这样对于答案是没有影响的。

至于这个东西怎么处理我们可以用并查集，但是并查集比较难求线段交，所以我们可以用线段树来实现。

然后欧拉定理可以发现，整张图的联通块个数就可以表示成交点个数加上联通块个数减去线段个数。

接下来唯一麻烦的就是如何求交点，这个可以直接扫描线轻松地解决。

时间复杂度 $O(n \log n)$。

「THUPC2018」好图计数

我们归纳定义一个无向简单图是**好的**：

1. 一个单点是好的。
2. 若干个好的图分别作为联通块所形成的图是好的。
3. 一个好的图的补图是好的。

给一个正整数n，求本质不同的无标号好图数量模质数P。

「THUPC2018」好图计数

我们归纳定义一个无向简单图是**好的**：

1. 一个单点是好的。
2. 若干个好的图分别作为联通块所形成的图是好的。
3. 一个好的图的补图是好的。

给一个正整数n，求本质不同的无标号好图数量模质数P。

$n <= 23333$

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。
我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。

我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

$$f_n = [x^n]\prod_{i=1}^{n-1}\left(\frac{1}{1-x^i}\right)^{f_i}$$

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。
我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

$$f_n = [x^n]\prod_{i=1}^{n-1}(\frac{1}{1-x^i})^{f_i}$$

我们取ln再求exp，得

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。
我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

$$f_n = [x^n] \prod_{i=1}^{n-1} (\frac{1}{1-x^i})^{f_i}$$

我们取ln再求exp，得

$$f_n = [x^n] e^{\sum_{i=1}^{n-1} f_i ln(\frac{1}{1-x^i})}$$

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。
我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

$$f_n = [x^n]\prod_{i=1}^{n-1}(\frac{1}{1-x^i})^{f_i}$$

我们取ln再求exp，得

$$f_n = [x^n]e^{\sum_{i=1}^{n-1} f_i ln(\frac{1}{1-x^i})}$$

其中$ln(\frac{1}{1-x^i}) = \sum_{j=1} \frac{x^{ij}}{j}$。

「THUPC2018」好图计数

我们考虑对于一个节点数为n的好图，我们把它分为连通和不连通两类，显然不连通好图唯一对应一个连通好图（经过补图转换）。
我们计数一共有多少节点数为n的不连通好图，它是由若干连通好图构成的集合，设i个点的连通好图一共f_i种，则

$$f_n = [x^n]\prod_{i=1}^{n-1}(\frac{1}{1-x^i})^{f_i}$$

我们取ln再求exp，得

$$f_n = [x^n]e^{\sum_{i=1}^{n-1} f_i ln(\frac{1}{1-x^i})}$$

其中$ln(\frac{1}{1-x^i}) = \sum_{j=1} \frac{x^{ij}}{j}$。
我们把式子展开即可分治FFT/牛顿迭代甚至$O(n^2)$递推。均可通过。

「THUWC2020」某科学的动态仙人掌

给定一棵n个节点的树和一个整数X，q组询问，每次询问给出l和r，将编号在l至r之间的点拿出来，如果其中有两点u, v满足它们在树上的距离 $\mathrm{dist}(u, v) \le X$，就在$u, v$间连一条边，问最终形成的连通块数量。

「THUWC2020」某科学的动态仙人掌

给定一棵n个节点的树和一个整数X，q组询问，每次询问给出l和r，将编号在l至r之间的点拿出来，如果其中有两点u,v满足它们在树上的距离 $\mathrm{dist}(u,v) \le X$，就在u,v间连一条边，问最终形成的连通块数量。

$1 \le n \le 3 \times 10^5, 1 \le q \le 6 \times 10^5$

「THUWC2020」某科学的动态仙人掌

我们定义一个点为一个连通块的根当且仅当它是该连通块的深度最小的节点，如果有多个那么要求编号也要最小，于是统计根的个数即可。

「THUWC2020」某科学的动态仙人掌

我们定义一个点为一个连通块的根当且仅当它是该连通块的深度最小的节点，如果有多个那么要求编号也要最小，于是统计根的个数即可。

然后我们看一个点在什么时候会成为根，我们找出一个编号小于它且深度小于等于它且距离小于等于X且编号尽可能大的点，和一个编号大于于它且深度小于它且距离小于等于X且编号尽可能小的点，这样如果我们选择的区间如果不包含这两个点，这个点就是根，我们对每一个点找出这二者，然后树状数组统计即可。

「THUWC2020」某科学的动态仙人掌

我们定义一个点为一个连通块的根当且仅当它是该连通块的深度最小的节点，如果有多个那么要求编号也要最小，于是统计根的个数即可。

然后我们看一个点在什么时候会成为根，我们找出一个编号小于它且深度小于等于它且距离小于等于X且编号尽可能大的点，和一个编号大于于它且深度小于它且距离小于等于X且编号尽可能小的点，这样如果我们选择的区间如果不包含这两个点，这个点就是根，我们对每一个点找出这二者，然后树状数组统计即可。

我们发现对于每一个点，需要搜寻的范围实际上是距离一个点或一条边距离为定值的点中一个值的前驱/后继。我们用动态点分治，线段树维护即可。

「CF739D」Recover a functional graph

给定一个n个点的基环内向树森林（每个点有且仅有一条出边）中每一个点到其所在基环内向树的环的最短距离和这个环的环长，其中有一些是 '?' 表示可以这个距离或环长可以由你决定。你需要构造一个边权均为1的图，给出一种决定所有 '?' 并给出所有点的连边的方案，使得给出的所有条件满足，或者说明不存在这样的方案。

「CF739D」Recover a functional graph

给定一个n个点的基环内向树森林（每个点有且仅有一条出边）中每一个点到其所在基环内向树的环的最短距离和这个环的环长，其中有一些是 '?' 表示可以这个距离或环长可以由你决定。你需要构造一个边权均为1的图，给出一种决定所有 '?' 并给出所有点的连边的方案，使得给出的所有条件满足，或者说明不存在这样的方案。

$1 \leq n \leq 300$

「CF739D」Recover a functional graph

显然一个环挂一条链是非常优的，只有对于一个环距离的某一层多出了一些点我们才会挂出树的形态

「CF739D」Recover a functional graph

显然一个环挂一条链是非常优的，只有对于一个环距离的某一层多出了一些点我们才会挂出树的形态

我们发现只有四种情况：

「CF739D」Recover a functional graph

显然一个环挂一条链是非常优的，只有对于一个环距离的某一层多出了一些点我们才会挂出树的形态

我们发现只有四种情况：

$x\ y$： 环长和距离是固定的，按照要求连。
$x\ ?$： 距离固定，先把所有环连出去最长边的空隙补完，有剩余再往最长边后接。
$?\ x$： 环长固定，直接往相应长度的环上补没有补完的距离空隙。

$?\ ?$： 都不固定，把剩下没补完的随便填，或者随便往后接。
如果全部用完了但还是有环不完整或者环连出去的边不完整就无解。

「CF739D」Recover a functional graph

显然一个环挂一条链是非常优的，只有对于一个环距离的某一层多出了一些点我们才会挂出树的形态

我们发现只有四种情况：

x y ： 环长和距离是固定的，按照要求连。
x ? ： 距离固定，先把所有环连出去最长边的空隙补完，有剩余再往最长边后接。
? x ： 环长固定，直接往相应长度的环上补没有补完的距离空隙。

? ? ： 都不固定，把剩下没补完的随便填，或者随便往后接。
如果全部用完了但还是有环不完整或者环连出去的边不完整就无解。

「CF739D」Recover a functional graph

其中由于x ?是填入多个环的空隙的，而? x只可以填补固定环的空隙，因此我们发现最优时应该在填完x ?后，剩余? x恰好能填入空隙。这样对于每一个环，我们可以留下? x数量的空隙由它来填，也就是说，每一个环用填入? x的空隙数必须大于等于某一个值，这样就是一个网络流模型了，跑最大流即可。

「CF1336E2」Chiori and Doll Picking (hard version)

给n个数a_i，$a_i < 2^m$，对于所有$i \in [0, m]$，求从其中选出一个子序列使得子序列中数异或和的二进制中1的个数等于i的方案数模998244353。

「CF1336E2」Chiori and Doll Picking (hard version)

给n个数a_i，$a_i < 2^m$，对于所有$i \in [0, m]$，求从其中选出一个子序列使得子序列中数异或和的二进制中1的个数等于i的方案数模998244353。

$1 \leq n \leq 200000, m \leq 53$

「CF1336E2」Chiori and Doll Picking (hard version)

首先我们可以求出线性基，然后就转化为求线性基中选一个子集使得异或和的$popcount$为i。

「CF1336E2」Chiori and Doll Picking (hard version)

首先我们可以求出线性基，然后就转化为求线性基中选一个子集使得异或和的$popcount$为i。

当线性基中元素个数$\leq B$时，我们可以暴力枚举，复杂度$O(2^B)$。

「CF1336E2」Chiori and Doll Picking (hard version)

首先我们可以求出线性基，然后就转化为求线性基中选一个子集使得异或和的$popcount$为i。

当线性基中元素个数$\leq B$时，我们可以暴力枚举，复杂度$O(2^B)$。

否则，我们考虑FWT，设最后异或出x的答案为A_x，线性基中数为$C_0, C_1, \ldots, C_{n-1}$，我们构造$P_{i,C_i} = P_{i,0} = 1$，那么有$FWT(A) = \prod_{i=0}^{n-1} FWT(P_i)$。

「CF1336E2」Chiori and Doll Picking (hard version)

首先我们可以求出线性基，然后就转化为求线性基中选一个子集使得异或和的$popcount$为i。

当线性基中元素个数$\leq B$时，我们可以暴力枚举，复杂度$O(2^B)$。

否则，我们考虑FWT，设最后异或出x的答案为A_x，线性基中数为$C_0, C_1, \dots, C_{n-1}$，我们构造$P_{i,C_i} = P_{i,0} = 1$，那么有$FWT(A) = \prod_{i=0}^{n-1} FWT(P_i)$。

容易发现A中只有0与2^n，考虑FWT的定义，即和某一位与的$popcount$为奇数则加上这一位的贡献，否则减去，因此$FWT(P_i)$只有0和2，所以得证。我们的证明同样给出了仅有当$\forall i, popcount(C_i \& j) \equiv 0 \pmod 2$时，$FWT(A)_j$为$2^n$。

「CF1336E2」Chiori and Doll Picking (hard version)

假设我们知道$FWT(A)$，那么我们就可以统计出$FWT(A)$中$popcount$等于i的位的和，然后用组合数得到答案，于是我们只要得到$FWT(A)$即可。

「CF1336E2」Chiori and Doll Picking (hard version)

假设我们知道$FWT(A)$，那么我们就可以统计出$FWT(A)$中$popcount$等于i的位的和，然后用组合数得到答案，于是我们只要得到$FWT(A)$即可。

我们发现根据以上条件$FWT(A) \neq 0$的位只有2^{m-n}个，因此我们可以暴力。

「CF1336E2」Chiori and Doll Picking (hard version)

假设我们知道$FWT(A)$，那么我们就可以统计出$FWT(A)$中$popcount$等于i的位的和，然后用组合数得到答案，于是我们只要得到$FWT(A)$即可。

我们发现根据以上条件$FWT(A) \neq 0$的位只有2^{m-n}个，因此我们可以暴力。

我们先把线性基消成这样的形式：其左边$n \times n$为单位矩阵，后$n \times (m-n)$是任意的。

「CF1336E2」Chiori and Doll Picking (hard version)

假设我们知道$FWT(A)$，那么我们就可以统计出$FWT(A)$中$popcount$等于i的位的和，然后用组合数得到答案，于是我们只要得到$FWT(A)$即可。

我们发现根据以上条件$FWT(A) \neq 0$的位只有2^{m-n}个，因此我们可以暴力。

我们先把线性基消成这样的形式：其左边$n \times n$为单位矩阵，后$n \times (m-n)$是任意的。

然后我们可以枚举后$m-n$位，在最后我们因为要使所有$popcount(C_i \& j) \equiv 0 \pmod 2$，我们可以唯一确定前$n$位，暴力的复杂度是$O(2^{m-n})$。

「CF1336E2」Chiori and Doll Picking (hard version)

假设我们知道$FWT(A)$，那么我们就可以统计出$FWT(A)$中$popcount$等于i的位的和，然后用组合数得到答案，于是我们只要得到$FWT(A)$即可。

我们发现根据以上条件$FWT(A) \neq 0$的位只有2^{m-n}个，因此我们可以暴力。

我们先把线性基消成这样的形式：其左边$n \times n$为单位矩阵，后$n \times (m-n)$是任意的。

然后我们可以枚举后$m-n$位，在最后我们因为要使所有$popcount(C_i \& j) \equiv 0 \pmod 2$，我们可以唯一确定前$n$位，暴力的复杂度是$O(2^{m-n})$。

这样取$B = 27$即可。

Thanks!