个人刷题

fks

CF1687C

题解:考虑转化题意，令c[i]=b[i]-a[i],再对c作前缀和，那么就转化为了，每次对于[l,r]如果c[l-1]=c[r],那么把l-r这段区间全部覆盖为c[l-1].要求我们最后能把所有c都变成0.我们倒着考虑，来看操作是否有用（因为一开始我对于区间两两交的影响很头痛）。考虑最后反正要都变成0.那么必然c[l]和c[r]也必须要是0.否则做了和没做一个样。那么我们只对是0的考虑。我们把操作存在两个端点的vector里。暴力判断和更新就好了。用set维护非0位置弹出□set用于均摊，只染色非0位置）。

CF1687D

题解：一开始想的，把段都暴力弄出来，后面扫描线做。但发现其实不需要，考虑可爱的区间是[k^2,k^2+k],那么我们就发现，当我们固定了a[1]的段，也就是a[1]的偏移量的范围确定，后面的段的偏移量也能唯一确定（因为一个段最多从可爱到不可爱，或者从不可爱到可爱，不可能跳变两次（凸性））。那么我们可以发现，一个在不可爱段的最小值，会对下界有影响。一个在可爱段的最大值，会对上界有影响。我们预处理出前驱后继，直接做。每次跳的次数是V/i□那么就是调和级数

CF1687E

题解：考虑给出的形式，实际上是每个因子的min和min_rk2相加，我们考虑他的选择方式，实际上暗示着我们考虑min_Max容斥，比较容易的可以得到式子（用广义minmax可以得到），然而我们发现复杂度是2^n*n□无法通过。似乎没法优化？我们换个方向，想想能否减小n□来简化问题。也就是说挑选出一些代表性的数，来与我们整个数列等价。我们考虑一个定理：一个数最多的因子个数是w(n)□在1e6内，这个函数是7。那也就是，我们可以每次钦定每个因子次小/最小给他选上。

CF1717E

□vp□ 题解：比较关键的一点就是想到枚举gcd□不枚举比较难做）。利用辗转相减，我们可以发现a=x*t,b=y*t,gcd(n-(x+y)*t,t)就变为了gcd(n,t)□但我们考虑要求出gcd□x,y□==1并且x+y=c的对数，考虑y=c-x,gcd(x,c-x)==1 ⇒gcd(x,c)=1,也就是欧拉函数

CF1717F

(vp) 题解：+1，-1比较难处理，考虑先全部-1，做单选题，每次选一个+2，那么就变成了考虑+2分配给谁的问题。我们做个delta□把目标值和当前值作差，考虑每个人要被加几次+2，也就是delta/2□ 那么可以判断掉一些不合法的□delta负、delta不能被2整除。 比较显然可以得到网络流模型，但我们考虑到有些是无限制，如果按照“ 最多 ”的约束连边，则能达到最大流，但没法满足恰好。考虑上下界网络流建图即可。

ljz

VP的签到题就不写了。

CF1730D

题意：每次可以把a的前k位和b的后k位交换，问a是否可以变成b□ 题解：容易发现，对于ai和bn-i+1这两个元素的对应关系不管怎么操作都是不会变的。所以有解无解的判断方法就是相同的pair□无序）有偶数个，或者有一个奇数且n是奇数。

CF1730E

题意：计算有多少个区间满足最大值可以整除最小值。 题解：先用单调栈求出以每个元素为最大值最小值的最长区间。我们对每个ai都求出以ai为最大值的满足条件的区间，枚举ai的每个约数□ai在一百万以内因此可以AlogA预处理），对于一个约数d□可以找到离ai最近的d□左右各一个），我们只需要计算包含ai和d的满足条件的区间，对于左右的d注意去重。

CF1730F

题意：给定一个排列p和k□找到一个排列q□满足任意i<j□都有pqi⇐pqj+k□使得q的逆序对数量最少□(n⇐5000,k⇐8) 题解：我们一个一个考虑q□对于q1□pq1的值只可能是[1,k+1]中的的一个，如果pq1取了1那pq2的值只可能是[1,k+1]中的一个，如果pq1没有取1那么pq2的值还是只能在[1,k+1]中取，后面以此类推，也就是说每放置一个q□我们需要考虑的值只有最小的没被选过的数到这个数+k的范围内，这样就可以状压DP了。

CF1733E

题意□n*m的方格图上每个格子有一个箭头，初始时(0,0)上有一个小球，以后每一个时刻所有小球都会按照箭头的方向走一个格子，然后(0,0)上会新出现一个小球，所有上一个时刻有球的格子都会转换箭头方向（只有向右和向下两种状态）。问第t时刻给定的格子是否有球□(t⇐1e18□x0y0⇐120) 题解：我们把所有小球统一考虑，也就是说一共会生成t个小球，小球走到(x0,y0)需要x0+y0-1个时刻，我们考虑有可能经过(x0,y0)的小球一共t-x0-y0+1个，如果经过当前格子的小球有n个，那么其中会有\left\lfloorn/\left.2\right\rfloor\right.个经过下方的格子，剩下的会经过右边的格子。我们只需要知道t时间经过(x0,y0)的小球数量和t-1时间经过(x0,y0)的小球数量是否相等就可以判断是否在t时刻(x0,y0)上有球。

CF1720D

题意：给定一个序列ai求ai的一个最长子序列bi□满足a_{b_p}\oplusb_{p+1}<a_{b_{p+1}}\oplusb_p对于任意p成立。题解：注意到两个数a<b在二进制意义下，一定是前一段二进制a=b□后一位上a是0□b是1。前一段二进制表示a_{b_p}\oplusb_{p+1}=a_{b_{p+1}}\oplusb_p\ \Rightarrow\ a_{b_p}\oplusb_p=a_{b_{p+1}}\oplusb_{p+1} 这样就可以用0/1trie维护a_i\oplus i□对于每个a_i都在trie中查找前面可以接的最长段。

CF1717E

题意：求\sum_{a+b+c=n} l c m\left(c,gcd\left(a,b\right)\right)其中abc都是正整数n\le300000 题解：我们枚举c则a+b=n-c\Rightarrow\ gcd(a,b)|n-c枚举n-c的约数d即gcd(a,b)=d问题转化为求a0+b0=\frac{n-c}{d},\left(a0,b0\right)=1的(a0,b0)对数，即\varphin-cd

CF1717F

题意：给定一张无向图，和一个点集S要给每条边定向，使得每个S中的结点i满足，入度-出度=一个给定值ain,m⇐10000 题解：对于规定的结点，需要有bi=(ai+度数)/2个入度，以此可以判掉一部分NO考虑网络流的做法，建立超级源点SS向每条边连一条流量为1的边，每条边分别向两个端点连一条流量为1的边，建立超级汇点T对于每个点集中规定的点，向T连一条流量为bi的边。跑一遍网络流，这个建图其实相当于一个二分图，因此网络流复杂度是比较优秀的。如果最终最大流和所有的bi的和相等则可能存在一组解，否则无解。但这样跑完网络流之后并不能保证每条边都被选择了一次，因为我们忽略了不在给定点集中的点，我们在上一次跑完网络流的图的基础上给每个不在给定点集中的点向T连一条流量为无穷的边，再跑一次网络流，如果两次最大流加起来是m才能保证有解。为什么不一开始就连？因为网络流很可能会优先把无穷的边都流满，这样构造的方案并不满足我们的条件。最终输出构造方案就看每条边的流向即可。

CF1716E

题意：有2^n个数，每次操作选定一个i对每个k交换a_kak+2i已经被交换过就跳过。有q次操作每次操作给定一个i求操作完的序列的最大子段和，每次操作的改变保留 q⇐200000,n⇐18 题解：如果下标从0开始，我们发现，每次就是a_kak⊕2i交换，每次操作的改变保留的话，就是把每个操作数异或起来得到一个xa_kak⊕x交换，我们可以对每个0⇐i<2^n都预处理出答案，而对于最大子段和，可以用线段树维护。对于每个i我们并不需要建出完整的线段树，我们可以利用i异或掉最高位的1之后的i′的线段树，这样总共只需要n·2n个结点。

CF1716F

题意：有n个口袋，每个口袋中有编好分别为1~m的m个球，从每个口袋中都拿出一个球来，设编号为奇数的球的个数为F求所有F^k的和，多组数据对998244353取模(n,m<998244353,k⇐2000,T⇐5000)题解：令a=\left\lfloor\frac{m+1}{2}\right\rfloor,b=\left\lfloor\frac{m}{2}\right\rfloor
则，ans=\sum_{i=1}^{n}\binom{n}{i}·ai·bn-i·ik =\sum_{i=1}^{n}\binom{n}{i}·ai·bn-i·j=1ksk,j·i!i-j! =j=1ksk,j·1j!·i=1nniij·ai·bn-i \ \ \ \ \ \ \ =j=1ksk,j·n!n-j!·i=1nn-ji-j·ai·bn-i =j=1ksk,j·n!n-j!·aj·(a+b)n-j

CF1715E

题意：有一张n个点m条有边权的边的无向图，求一个人从1出发到达各个结点的最短用时，任意两个结点u,v之间可以乘坐飞机，用时{(u-v)}^2,最多坐k次飞机(n,m⇐100000,k⇐20) 题解：考虑dpi,j表示恰好坐j次飞机到i号点的最短用时，状态由dpk,j-1转移过来，是很标准的斜率优化形式，但是要注意这样转移完成后dpi,j表示的是恰好坐j次飞机到i号点并且最后一步是坐飞机抵达i的，因此还要用迪杰斯特拉跑一遍dpi,j

CF1715F

题意：交互，有一个n*m的网格，有一个1*1的方格需要你确定他的位置，每次你可以询问给出一个多边形，交互器返回此多边形和那个1*1交的面积，最多询问5次n,m⇐100 题解IQtest考虑锯齿状的多边形， 这种多边形可以根据面积确定一维坐标，其实只需要询问两次就行了。

CF1713E

题意：有一个n*n的矩阵，每次操作可以选择一个k把矩阵的第k行和第k列转置，问通过操作可以得到的最小字典序的矩阵。 题解：对于a_{i,j}aj,i如果要交换，只能选择i或者j操作，对于1⇐i⇐n只有两种状态，操作或者不操作，因为对于同一个位置操作多次是抵消的，而且操作顺序与最终答案无关。按顺序枚举a_{i,j}<a_{j,i}则ij的状态应该是相同的，反之ij的状态应该不相同。可以用带权并查集或者种类并查集维护这些关系。同时还有一个观察，就是对于一种可行的操作，操作他的补集后的结果也是一样的。因此对于一个并查集的根随便选择操作或不操作都可。

CF1712E

题意T次询问，每次给了l,r求满足lcm(i,j,k)\geqi+j+k,l\lei<j<k\ler的对数T⇐100000,l,r⇐200000题解：我们发现满足条件的三元组数量应该是很多的，如果lcm(i,j,k)\geq3*k那么肯定满足条件，所以可以考虑反面，因此我们只需要计算lcm\left(i,j,k\right)=klcm(i,j,k)=2*k的三元组数量。首先考虑lcm(i,j,k)=2*k的情况，还需要满足k<i+j其中ij都是2k的约数，这种情况下只有两种可能i=\frac{1}{2}k,j=\frac{2}{3}k或者i=\frac{2}{5}k,j=\frac{2}{3}k这都很好统计。接下来考虑lcm\left(i,j,k\right)=k那么只要满足ij是k的约数。预处理处每个数的约数，离线把询问按右端点排序，按顺序计算i<j<k\ler的三元组数量，回答询问时减去i<l的三元组，这可以用树状数组维护。

CF1709E

CF1706D

CF1697E

CF1696F

CF1696G

ARC149D

ARC149E

ARC149F

ABC271G

ARC148E

个人学习

ddp

新学了ddp□目前只做了P4719□还未学全局平衡二叉树□ddp实际上就是维护了一段重链的转移矩阵，难点在于矩阵的设计和dp状态的设计，可能有时候dp状态会为了转移，把一些权值给并入，比较抽象。需要维护轻儿子的dp值和当前的dp□每次更新一条重链的权值，再把更新重链顶端的父亲的权值。由于每个点维护了轻儿子的dp□且转移矩阵只和轻儿子的dp有关，所以每次修改的矩阵数只有log

updata(2022.10.12)□有了点新的理解，全局平衡二叉树感觉就是□lct用加权重心建树的方式，把每个平衡树建出来（权重是一个节点对应轻子树的大小）。然后操作就和lct一样□access的时候不需要splay□只需要上传信息即可