杂题选讲

kcz

Big Secret

- 给你一个长度为n的数组b，请你重新排列b，使得前缀异或和递增。
- n<=1e5,1<=b[i]<2^60

Source：CF925C

Big Secret

- a^x>a 等价于 a在x的最高位为0。
- 首先考虑最高位最大的那个数mx，
- 如果有多个数最高位也是这个位，那么无解；
- 接下来考虑下一位，
- 如果mx这一位为0，则允许1个，否则允许2个；
- 同理，考虑第i位时，第i位为1的数有x个，则允许x+1个数最高位为i。
- 时间O(nlog_A)

Nice table

- 给你一个n*m的矩阵(table)，每个格子是A,C,G,T中的一个。
- 定义一个table为nice table，当且仅当它的每个2*2的子矩阵都含有四种不同字符。
- 请你改变尽量少的格子，使得给定table变成nice table。
- n*m<=3e5

Source：CF1098C

Nice table

- 对于一个nice table，要么每行只有两种字符，要么每列只有两种字符，暴力枚举即可。

最长公共子序列

- 对于两个给定的序列，请求出它们的最长公共子序列长度。
- |s1|,|s2|,s1_i,s2_i<=70000

Source：LOJ

最长公共子序列

- 令dp[i][j]=lcs(s1_{1..i},s2_{1..j})
- 考虑对dp[i][j]差分，然后bitset优化。

最长公共子序列

- 具体来说，令f[i][j]=dp[i][j]-dp[i][j-1],b[c][i]=s2[i]==c
- 考虑从f[i-1]转移到f[i]的过程
- 把f[i-1]和b[s1[i]]按从高到低的顺序从左到右写出来
- 对于f[i-1]的每一段100000，f[i]的1都会移动到b的最右的1(如果不存在就不变)。(假装在f[i]的无穷高位始终存在1)
- 例如:
- f[i-1]= 000000010001
- b[s1[i]]= 010110001000
- 则f[i]= 000010001001

最长公共子序列

- 考虑如何转化成能用bitset优化的操作。
- 令X=f[i-1]|b
- 考虑y=X-((f[i-1]<<1)+1)
- 这相当于对于X的每一段，最末的10000变成了01111
- 考虑y^X
- 这相当于对于X的每一段，最末的10000变成了11111，而其他的1都变成了0
- 再跟X取and即可。
- 所以f[i]=((X-((f[i-1]<<1)+1))^X)&X
- 参考：IOI2006集训队作业《基于位运算的最长公共子序列》唐文斌

Cool Slogans

- 给你一个字符串T
- 求一个字符串序列S_1,S_2...S_K，使得S_1是T的子串,且对于所有2<=i<=K，S_i在S_{i-1}中出现了至少两次。
- 求最大的K。
- |T|<=2e5

Source：CF700E

Cool Slogans

- 对$S_{1..K}$进行一些变换，可以使得S_K长度为1，且$S_{i+1}(i<K)$在S_i中出现恰好两次，且一次是前缀，一次是后缀。
- 定义一个串S是好串，当且仅当S长度为1，或者存在好串T满足T在S中出现恰好两次，且一次是前缀，一次是后缀，这时称T是S的父亲。
- 则存在$S_{1..K}$使得K取到最大值，满足S_i都是好串，且$S_{i+1}(i<k)$是S_i的父亲。

Cool Slogans

- 引理：对于一个好串S，设它所有的好的前缀分别为T_{1..k}，则T_{i-1}(i>1)是T_i的唯一的父亲。
- 可以归纳证明。
- 从右到左枚举好串的左端点，则每次会恰好发现一个新的好串。
- 因为对于一个串S_{1..n}，设所有好的前缀为T_{1..k}，则T_{i-1}(i<k)是T_i的父亲，因此T_{i-1}在S_{2..n}中出现过。因此有且仅有T_k没在S_{2..n}中出现过。

Cool Slogans

- 考虑怎么求出这个新的好串T_k。
- 实际上它就是S_{i..n}最长的好的前缀。
- 设l_i=|T_k|,
- 则|T_{k-1}|=max{l_j | lcp(i,j)>=l_j},
- 若|T_{k-1}|=0，则l_i=1;
- 否则l_i=|T_{k-1}|+min{j | lcp(i,j)>=|T_{k-1}|}-i。
- 可以用后缀数组来优化转移。
- 答案顺便dp一下就好了。
- 时间复杂度O(nlogn)。

Mateusz and an Infinite Sequence

- 给你一个长度为d的数组gen和模数m，
- 定义序列M_k如下：
- $M_0=\{0\}$
- $M_k(k>=1)$是将M_{k-1}复制d份，且复制的第i份的所有元素都要加上gen_i。
- 保证$gen_1=0$，给你一个序列B和两个数l,r，问B在M_{∞}的$[l,r]$区

间共出现了多少次。

- $2<=d<=20, 2<=m<=60, |B|<=30000, 1<=l<=r<=1e18$

Source：Hello2019

Mateusz and an Infinite Sequence

- 对于一个M_{oo}的区间，我们维护以下信息：
- 长度len，B的出现次数ans，
- bitset pre,suf，pre[i]表示其长度为i的后缀能否和B长度为i的前缀匹配(不足的地方认为可以匹配)，suf[i]表示其长度为i的前缀能否和B长度为i的后缀匹配。
- 则可以在O(|B|/32)的时间内合并两个区间的信息。

Mateusz and an Infinite Sequence

- 下面类似数位dp搞一下就好了。
- dp[i][v]表示M_0={v}时M_i的信息。
- 时间复杂度O(log(r)*m*d*|B|/32)。

Manju Game

- 给你一个序列a_{1..n}，A和B交替操作，A先手。
- 每次操作如下：
- 1.任意选择一个还未被选择且和对方最后一次选择相邻的元素。
- 2.如果不存在满足1.的条件的元素，或者是A的第一次选择，则任意选择一个还未被选择的元素。
- 当所有元素都被选择时，游戏结束。
- A和B都会最大化自己选择的元素的和，求A和B选择的元素的和。
- n<=3e5,a_i<=1000

Source：AGC026F

Manju Game

- 将奇数位置的颜色记作B，偶数位置的颜色记作W。
- 1) n%2==0
- 如果先手选了一个W，且不是最右边的W
- 序列被分成一奇一偶两半，
- 此时如果后手选right，则左半边变成后手先选，此时如果后手选最右边的B，则可以带走所有B。
- 故先手不如选最右边的W。
- 因此ans=max(sum(B),sum(W))

Manju Game

- 2) n%2==1
- 如果先手选B，同理必须选最边上的B，此时ans=sum(B)。
- 如果先手选W，序列会被分成长度都为奇数的两半，后手需要选left or right，然后另一半递归。
- 如果先手的策略为：先选b，若后手选right，则先手选a，
- 那么可以将策略改为：先选a，若后手选left，则先手选b。
- 因此先手的策略中选择的W可以变成递增，即可以变成这样的形式：
- 先选a,
- 若后手选right,则先手带走left的B，然后结束；
- 否则若后手选left，则先手选b然后递归。

Manju Game

- 这时可以直接dp，数据结构优化做到$O(n \log n)$，不过有更好写的方法。
- 对于策略序列$x_{1..k}$，k+1段中一定存在恰好一段先手选的是B，其他是W，
- 最差情况一定是k+1段中sum(B)-sum(W)最小的那一段。
- 设ans=sum(W)+X，则要求min{sum(B)-sum(W)}>=X；
- 二分X，O(n) dp。
- 时间$O(n\log{\sum_a})$

Coloring Torus

- 给你一个整数K，
- 请你构造一个n*n的网格，给每个格子选择[1..K]中的一个颜色，满足：
- 1.所有K种颜色都出现过。
- 2.对于任意颜色i,j，所有颜色为i的格子颜色为j的相邻格子数相同。
- 这里第1行/列和第n行/列相邻，因此每个格子都有恰好4个相邻的格子。
- K<=1000,n<=500

Source：AGC030C

Coloring Torus

- 考虑n*n的网格的n条斜线，每个点的相邻点即上面一条斜线的两个点和下面一条斜线的两个点。
- 令n为偶数，对于每条斜线，要么只用一种颜色，要么用两种颜色交替，这样可以做到[n,2n]种颜色。

Construction of a tree

- 给你n-1个{1..n}的子集，令第i个为E_i。
- 请你在每个E_i中选出两个元素u_i,v_i，使得所有(u_i,v_i)构成一棵树。
- 无解输出-1。
- n<=1e5,\sum |E_i|<=2e5

Source：AGC029F

Construction of a tree

- 考虑以1为根时，n-1条边就形如(u,p_u)；
- 建一张二分图，左边的a和右边的b有边当且仅当a \in E_b；
- 网络流求出(2..n)和E_{1..n-1}的一个完美匹配，
- 如果不存在完美匹配显然无解；
- 然后将1放入队列，每次取出队头x，对于所有x \in E_i且没访问过的i，将i的匹配点记作match[i]，令p[match[i]]=x，并将match[i]入队。
- 这样如果最后存在没有被入队的点，说明二分图一开始就不连通，那么显然无解。

constructive

- 给你一个整数n，
- 请你构造一个n*n的矩阵A，满足A的每行每列都是1..n的一个排列，且对于任意i<j，A_{i,j}!=A_{j,i}。
- 无解输出-1。
- n<=1000

constructive

- 首先考虑这样一个构造：首先令对角线上都是1，然后把每个1的右边置为2，2的右边置为3。。
- 可以发现，这样在n为奇数时满足要求的。

constructive

- 下面考虑n为偶数的情况。
- 当n=2时，显然无解。
- 当n=4时，通过爆搜发现有解。

constructive

- 对于一般的n，考虑将n*n的矩阵分成4个(n/2)*(n/2)的子矩阵，
- 令[1,n/2]放在左上和右下的子矩阵，[n/2+1,n]放在右上和左下的子矩阵。
- 对于[1,n/2]只用递归构造即可(当n=4时不能递归，需要特判)；
- 对于[n/2+1,n]，
- 先随便构造出左下角那个子矩阵(比如用n为奇数时那个构造)，
- 然后先令右上角的子矩阵满足a[i][j]=a[j][i]，即左下角子矩阵的转置矩阵，
- 然后将这个矩阵循环右移一位，这样当n/2>1时显然满足要求。
- 因此当且仅当n=2时无解。

子序列

- 给你一个仅含小写字母的字符串t_{1..n}
- 这样定义一个字符串序列s_{1..n}
- s_0为空
- s_i由在s_{i-1}的每个字符之间以及开头,结尾插入t_i得到。
- 求s_n的本质不同的子序列个数，mod 998244353。
- n<=2000

子序列

- 考虑这样一个操作：$s_i=s_{i-1}cs_{i-1}$
- 对$t_{1..n}$顺序执行原操作等价于对$t_{n..1}$顺序执行新操作。
- 考虑如何对一个字符串的子序列进行计数，建个自动机转化成路径计数。
- 考虑如何合并两个字符串，dp[i][j]表示第一条边为i，即将走j(j=0表示结束)即可，合并等价于矩阵乘法。
- $O(m^3n)$。

Immortal ... Universe

- 有仅由P和V组成的两个字符串A,B。其中P表示选了之后会亏1元，V表示选了之后会赚1元。
- 男孩一开始有1元。他每次会随机选择其中一个非空字符串，取走其开头字符。当男孩没钱的时候，他就破产了。
- 然而，当他只有1元的时候，他会得知两个字符串的开头字符(如果存在的话)，这时他会贪心，能选V就随机选一个V。当然，如果此时只能选P，那他就破产了。
- 现在给你两个仅由P,V,?组成的字符串A,B，问在把每个?替换成P或V的所有情况中，有多少种情况男孩一定不会破产。模998244353
- $|A|,|B|<=5000$

Source：EC final

Immortal ... Universe

- 考虑什么情况下男孩可能破产。
- 当男孩下一步就要破产时，假设A取到了i，B取到了j，则A[i]=P或i>|A|，且B[j]=P或j>|B|，且sum(A[1..i-1])+sum(B[1..j-1])=0。
- 同时，反过来，如果存在i,j，满足A[i]=P或i>|A|，且B[j]=P或j>|B|，**且不能i>|A|的同时j>|B|**,且sum(A[1..i-1])+sum(B[1..j-1])=0，那么男孩就可能破产。

Immortal ... Universe

- 这就等价于$\min\{sum(A[1..i-1])|A[i]=P$或$i>|A|\}+\min\{sum(B[1..j-1])|B[j]=P$或$j>|B|\}<=0$
- (先不考虑$i>|A|$且$j>|B|$的情况)
- 因此可以对A,B分别求出最小值=j的方案数，然后枚举A的j和B的j计算答案。
- 直接dp，dp[i][sum][mn]，$O(n^3)$。

Immortal ... Universe

- 如果给定A，这个最小值可以倒着求，只用一个变量nmn。
- dp[i][nmn]，O(n^2)。

Dev, Please Add This!

- 给你一个n*m的网格，每个格子可能是一个墙，一个空格，一个球或者一颗星。
- 有且仅有一个格子上是球。
- 你可以移动球，每次指定一个方向，球就会一直滚直到下一个格子为墙或者在网格之外。
- 当球经过一颗星时，就会收集到这颗星。
- 请你判断能否收集到所有的星。
- n,m<=50

Dev, Please Add This!

- 横条竖条拎出来建图，缩点变成DAG，然后问题变成求给定起点的一条路径，使得满足给定限制，每个限制为两个点，要求至少经过一个。

Dev, Please Add This!

- 2-sat
- 对每个点，设布尔变量表示路径是否经过这个点。
- 1 对于每个限制，两个点不能都是0。
- 2 对于起点不能到达的点，=0。
- 3 对于互相不能到达的点对，不能都是1。
- 满足条件的路径一定满足这些限制；
- 对于满足这些限制的布尔变量取值，一定可以构造出一条路径。
- O(n^4)

讲完了

祝大家身体健康