集训队作业Solution

中山纪念中学：曹天佑

November 1, 2019

CF526F Pudding Monsters

Description:
在一个n*n的网格图上有n个特殊格子，保证每行每列都恰好有一个特殊格子
对于一个k*k($1 \leq k \leq n$)的子网格图，我们称其为好的，当且仅当其内部有恰好k个特殊格子
问好的子网格图的数量
$1 \leq n \leq 3 * 10^5$

Analysis:
对于所有的特殊格子(x_i, y_i)，由于x_i, y_i互不相同，我们将所有格子按x_i排序，所得到的y_i是一个排列
那么任何一个k*k的好的子网格图都可以用一个长度为k的满足这个区间内部的值域是连续的区间表示
问题变成了求给定排列的连续段个数

CF526F Pudding Monsters

进一步的，我们考虑怎样的区间是连续段
由于序列是排列，设max为区间最大值，min为区间最小值，我们很容易得出区间[l,r]为连续段的条件是$r-l+1=max-min+1$
于是我们现在要求的就是，有多少个区间[l,r]满足$r-l=max-min$

Solution 1:
考虑分治，我们求跨过分治中心的区间个数
那么我们可以讨论max和min在分治中心的哪一侧
先考虑max和min在同一侧，例如左侧的情况，右侧同理
枚举了l之后，我们发现我们已经知道了max, min, l，那么$r = max - min + l$可以直接计算得出，直接判断一下这个r是否合法即可
再考虑max和min在不同侧，例如max在左侧，min在右侧
那么我们枚举l，那么我们要
求$max[mid+1, r] < max[l, mid]$，这样的r是一段前缀
同时我们还要求$min[mid+1, r] < min[l, mid]$这样的r是一段后缀
也就是说，确定了l之后，合法的r是一段区间
我们要求一段区间中有多少个$min + r = max + l$，用一个桶维护一下即可
复杂度$O(n \log n)$

Solution 2:
我们考虑枚举r，那么我们相当于要问有多少个l满足$max - min + l - r = 0$
我们对每个l实时维护这个值，考虑右端点右移时的变化：
r的变化相当于是区间-1
l没有变化
max和min的变化都是一段后缀
我们预处理出max和min的变化范围，那么上面的操作都可以用线段树区间加来维护
注意到$max - min + l - r$一定≥ 0，问区间$= 0$的数的数量可以直接用区间最小值来维护
复杂度$O(n \log n)$

Solution 3:
关于连续段的问题大部分都可以直接用析合树来解决
对于一个析点，其单独为一个连续段
对于一个合点，其所有连续的儿子都为连续段
直接建出析合树统计一下即可
复杂度$O(n)$

Description:
给出一个n*m的矩阵A，满足A中的元素为1到n*m的排列
现在将A中的每一行，将其中的元素任意重排，得到矩阵B
现在将B中的每一列，将其中的元素任意重排，得到矩阵C
现在将C中的每一行，将其中的元素任意重排，得到矩阵D
要求D[i][j]=(i-1)*m+j，请给出一种方案
$1 \leq n, m \leq 100$

Analysis:
我们倒过来考虑所有操作，我们需要知道要满足条件，矩阵需要是什么样子的
在第二次操作之后，矩阵C的第i行必须要为一个(i-1)m+1,....,..im的排列
我们可以令这些数的颜色为i，那么这个条件等价于，在第二次操作之后，矩阵C的第i行的颜色必须全部为i

那么在第二次操作的时候，我们可以将所有的列重排，我们希望将颜色为1的格子放到第1行，颜色为2的格子放到第2行...
也就是说，我们希望每一列都包含了每种颜色，且每种颜色恰好出现一次

发现我们只需要考虑第一次操作，我们希望让每一列包含所有颜色

那么我们的问题变为了：给出一个n*m的矩阵，有n种颜色，每种颜色恰有m个格子，你可以重新排列每一行的格子，使得每一列包含所有颜色

我们按列考虑，每次给一列填上颜色
我们可以构造如下的二分图：
二分图的左部包含n个点，分别表示第1行到第n行
二分图的右部包含n个点，分别表示颜色1到颜色n
对于一个在第i行颜色为j的格子，从左部的第i个点向右部的第j个点连边
那么一种填某一列的方案对应这张图的一个完美匹配
我们找到这张图的一个完美匹配，如果第i行匹配了颜色j，那么我们将第一列的第i行填上颜色j
我们对每列重复以上操作，每次把匹配边删去，继续找完美匹配

Proof:
现在我们需要证明，对于这样的二分图，我们一直会有完美匹配
我们会发现，无论何时，每种颜色剩余的数量和每一行剩余的格子数都是相等的
也就是说，二分图中2n个点的度数都是相同的,设所有点的度数为m
考虑反证，假设二分图不存在完美匹配
那么根据Hall定理，我们会存在左边点的一个子集S,使得在右边和S集相邻的点数$< |S|$
设$|S| = k$，右边和S集相邻的点数为p,那么左边点的度数和为km,即右边点和左边点相连的边数为km
又$p < k$，则必然存在一个点，与其相邻的左部点的个数$> m$,这样就与所有点的度数都为m的题设不符

由此知道我们只需要每次取出一组完美匹配边，将其删去，最后一定能把所有边删完

用dinic做二分图匹配的复杂度是$O(nm\sqrt{n})$，总复杂度为$O(nm^2\sqrt{n})$

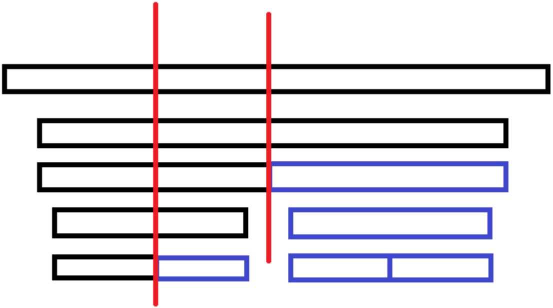
Description:
对于一个01串，我们可以将其编码，方式如下：
字符串'0'和'1'分别可以被编码为'0'和'1'
如果字符串A可以被编码为P，B可以被编码为Q，那么AB可以被编码为PQ
如果字符串A可以被编码为P，对于$k \geq 2$，字符串AAAA....(k个A)可以被编码为(Pxk)
例如，字符串001001001可以被编码为001001001,00(1(0x2)x2)1,(001x3)等等
我们称字符串A是字符串B的子集，当且仅当，对于所有的i满足A[i]=1，同样会有B[i]=1
现在给出字符串S，求所有S的子集的编码方案数之和，答案对998244353取模
$1 \leq |S| \leq 100$

Analysis:
面对这样的数据范围，我们显然不能对S的每个子集分别求编码方案数，我们需要把S的子集放在一起计算
令$f(S)$表示字符串S的所有子集的编码方案数之和，考虑S中的第一个字符的编码方案，我们有：
1:单独编码。那么剩下的字符串就有$f(S[2..|S|])$种方案。因为我们计的是子集，所以若$S[1]=1$那么还需要乘上2
2:被编码成(Pxk)。这里P是某个字符串A的编码，那么我们需要AA...A(k个A)这个字符串是$S[1..k|A|]$的子集
这个东西等价于，A是$S[1..|A|]$的子集，A是$S[|A|+1..2|A|]$的子集，...，A是$S[(k-1)|A|+1,k|A|]$的子集

我们定义两个长度相等的字符串P和Q的and(∧)是一个字符串，
满足$(P \wedge Q)[i] = min(P[i], Q[i])$
那么我们知道A是$S[1..|A|] \wedge S[|A|+1..2|A|] \wedge ...$
$\wedge S[(k-1)|A|+1, k|A|]$的子集
即枚举了$|A|$和k后，这部分可以从$f(S[1..|A|] \wedge S[|A|+1..2|A|] \wedge$
$... \wedge S[(k-1)|A|+1, k|A|]) * f(S[k|A|+1..|S|])$转移过来

这样子做的状态数看上去是$O(2^n)$的，但这个做法足以通过本题
实际上，通过实践可以知道有用的S的数量不超过50000个

Proof:
我们考虑哪些字符串T会被计算到，我们可以通过以下两种操作得到字符串T
1:取一个子串
2: “k-fold” 这个字符串，即从$S[1..k|A|]$得到A
当$|T| > N/8$时，我们最多执行两次fold，例如：先取一个子串，再2-fold，再取一个子串，再2-fold
剩余的情况是类似的



如图所示，我们在红线的地方fold，最后的串是所有蓝色串的$and(\wedge)$
我们只需要确定两个蓝色串的开头以及长度就能够确定T
这样就得到了一个上界$O(n^3+2^{n/8})$，实际上这个上界还是很松