题目选讲

NOIP 2017 模拟赛　神奇的三角形

- 在高三某一节数学课上，数学老师问到了杨辉三角的前 n 行的和。
- 郭神当然知道答案是 2^n-1，但他认为这个问题实在是 too young too simple，于是他把三角形的第 i列(最左边所有 的 1 为第一列，第二列为 1, 2, 3, 4, 5, 6以此类推) 乘上了 i^m，再询问前 n 行的和模 998244353，并把这个问题丢给了 SD_le。
- SD_le 的智商实在是捉急，并不能做出来，他只好向机智的你求助了。

- 对于 95% 的数据：m≤3000。
- 对于 100% 的数据： n≤10^9,*m*≤100000。

- 40%
- 首先有一个等式
- $\sum(0<=i<=n-1)C(i,x)=C(n,x+1)$
- 所以可以把问题转换到第n+1行上，相当于求$\sum(1<=i<=n)C(n,i)*i^m$
- 复杂度O(nlogm)
- 100%
- 考虑这个式子的组合意义
- 相当于先枚举i，然后在n个颜色里选出i个颜色，再用这i个颜色可重复的排出一个长度为m的排列的方案数。
- 换个思路。
- 枚举最后排列中实际用了几种颜色(*C(n,i))，其他的颜色可以选或不选(*2^n-i),然后再乘上i种颜色能排出的排列数，且每种颜色必须出现，就是S(m,i)*i!,S为第二类斯特林数，表示把m个物品放入i个相同盒子且盒子非空的方案数，因为本题是一个排列，所以需要乘上i！。
- 第二类斯特林数可以用dp或容斥求，复杂度O(n^2).
- 将容斥的式子化简一下，发现是一个卷积的形式，用NTT可以求出S(m,1)到S(m,m)，复杂度O(mlogm+mlogn).

BZOJ 4028: [HEOI2015]公约数数列

- 设计一个数据结构. 给定一个正整数数列 $a_0, a_1, ..., a_{n-1}$，你需要支持以下两种操作：
- 1. MODIFY id x: 将 a_{id} 修改为 x.
- 2. QUERY x: 求最小的整数 p ($0 <= p < n$)， 使得 $gcd(a_0, a_1, ..., a_p) * XOR(a_0, a_1, ..., a_p) = x$. 其中 $XOR(a_0, a_1, ..., a_p)$ 代表 $a_0, a_1, ..., a_p$ 的异或和，gcd表示最大公约数。

- 对于 100% 的数据，$n <= 100000$，$q <= 10000$，$a_i <= 10^9$ ($0 <= i < n$)， QUERY x 中的 $x <= 10^{18}$，MODIFY id x 中的 $0 <= id < n$， $1 <= x <= 10^9$.

- 观察到0~x的gcd最多变化log次，因为它每次变化一定至少要去掉一个质因子，所以我们可以枚举gcd。
- 因为数据范围比较小，所以想到了分块。
- 设T为块的大小。
- 维护块首到块里每个位置的gcd和xor，再把xor排序。
- 修改的时候暴力改就行，复杂度TlogT。
- 询问的时候如果gcd在这个块里变化了，就把这个块暴力扫一遍，否则说明gcd在这个块里不变，相当于在区间里查是否有某个特定的值，随便二分一下，复杂度Tloginf+nTlogT。

51NOD 二分答案

- lyk最近在研究二分答案类的问题。对于一个有n个互不相同的数且从小到大的正整数数列a（其中最大值不超过n），若要找一个在a中出现过的数字m，一个正确的二分程序是这样子的：
- l=1; r=n; mid=(l+r)/2;
- while (l<=r)
- {
-     if (a[mid]<=m) l=mid+1; else r=mid-1;
-     mid=(l+r)/2;
- }
- 最终a[r]一定等于m。
- 但是这个和谐的程序被熊孩子打乱了。
- 熊孩子在一开始就将a数组打乱顺序。（共有n!种可能）
- lyk想知道最终r=k的期望。
- 由于小数点非常麻烦，所以你只需输出将答案乘以n!后对1000000007取模就可以了。
- m<=n<=10^9

- 题目中的期望*n!可以转化成存在多少方案，使得r=k。
- 要使得r=k，我们仍可以通过二分来解决这个问题。假设此时mid>k，则必须存在a[mid]>m，否则必须存在a[mid]<=m。
- 那么题目就相当于了存在logn个这样的限制的数。
- 对于所有限制，我们可以通过乘法原理将它们的方案总数算出来。
- 对于没有限制的数，它们的方案总数其实就是一个数的阶乘。
- 这个阶乘我们可以通过打表求出来。
- 在程序中打出(1e7)!,(2e7)!,...,(1e9)!。每次在计算时只需再计算不超过1e7次就可以了。

51NOD　完美序列

- 如果一个序列的相邻两项差的绝对值小于等于1，那么我们说这个序列是完美的。给出一个有序数列A，求有多少种完美序列排序后和数列A相同。

- 第一行一个数n（<=30000）表示完美序列的长度
- 第二行n个数，表示数列A（每个数<=10^9，每个数出现次数<=100）

- A数组相邻的两个数差的绝对值也必须小于等于<=1，否则输出0设m为数值种数考虑依次插入每一种数值
- 1 1 2 2 2 1 1 2
- 这个例子中3只能插入到两个2中间或者最后一个2之后，插入之后会形成
- 1 1 2 2 3 3 3 2 1 1 2 3 3
- dp[i][j][0/1][0/1]表示处理完前i种数值，有j对相邻的数都为i，最前面和最后面是否为i，那么通过枚举i+1插入到几对i~i之间和是否插入最前面和最后面即可完成转移
- 显然有用的状态数最多就n*2*2种
- 时间复杂度O（n*100*16）

51NOD 最短路

- lyk有一个01矩阵，它一开始在(1,1)处，它想走到(n,m)，不幸的是它被剥夺了向左走的能力，也就是说，若lyk处于(i,j)，那么每一次lyk只能走向(i+1,j),(i,j+1),(i-1,j)三个地方，它所耗费的时间为它走过的路径中1的数量。它想知道从(1,1)走到(n,m)的最短路是多少。
- 但是这个问题十分的简单，lyk并不屑知道。
- 现在问题来了，lyk不想告诉你01矩阵是啥，它想知道的是从(1,1)走到(n,m)最短路为k的概率。
- 因此你需要输出的是当n+m行，第i行表示最短路为i-1的概率。
- 由于要输出小数点十分麻烦，因此你只需要输出概率*2^(n*m)后对p取模后的答案就行了。
- Input
- 一行3个数n,m,p。(2<=n<=6,1<=m<=100,1<=p<=1e9)。

- 问题的关键在于只能向右上下走，因此对于同一列，相邻两个位置的最短路一定不超过1。因此可以状态压缩DP。枚举第一行第i列的最短路是多少，再用3进制来表示下面第2行到第n行的最短路，就可以表示成这一列的最短路了。
- 预处理出对于所有3进制，且假设第一行那一列的最短路为0，枚举下一列的所有2^n状态时下一列的最短路。
- 这样我们在dp的时候，就是可以直接使用预处理出来的数组。
- 具体的，我们可以这么做。
- 令dp[i][j][k]表示当前枚举到第i列，第一行第i列的最短路为j，第2行~第n行的最短路的三进制以k来表示。
- 然后这个状态可以转移到所有我们刚才预处理出的2^n个状态里去。
- 这样状态时3^n*m^2，转移是2^n。
- 因此总复杂度为6^n*m^2

51NOD 调查任务

- bn是战忽中心——一个绝密的军事组织的一个军官，今天他接到了一个紧急任务：调查敌国X国某些城市的经济情况。
  X国有**N**个城市，由**M**条单向道路连接，其中**S**城是X国的首都。每个城市i有一个发达指数**a[i]**，我们定义城市i的经济状况为首都S到城市i任意一条路径上两个不同的城市x,y的a[x] mod a[y]的最大值。（**x**和**y**必须在同一条路径上，**x**，**y**可以是**i**或者**S**）
- Input
- 第一行四个正整数N,M,Q,S
- 分别表示X国城市数量，城市间有向边的数量，需要调查的城市的数目和首都的编号。每个城市的标号为1到N
- 第二行N个正整数，其中第i个整数表示a[i]。
- 第2至M+1行每行两个正整数x,y。表示有一条从城市x到y有向边。
- 第M+2行Q个正整数，表示需要调查的城市的数目的编号。
- 数据保证无重边无自环，不会查询首都。
- 1<=N,Q<=4*10^5
- 1<=M<=2*10^6
- 1<=a[i]<=10^9

- 首先我们考虑如果给你一个序列 {a[i]}，求 a[x]%a[y] 的最大值。

如果 a[x]>a[y]，那么 a[x]%a[y]<a[y]
如果 a[x]=a[y]，那么 a[x]%a[y]=0
如果 a[x]<a[y]，那么 a[x]%a[y]=a[x]

因此可以得到 a[x]%a[y]<=min(a[x],a[y])，那么对于每一组 (a[x],a[y])，如果两者不相等，则答案可以取到较小值，那么全局的答案也可以很容易得到：序列中的严格次小值。

我们考虑在这道题中的情况。首先访问的点次数不限，因此一个强连通分量中的点全都可以访问，那么我们缩点以后令每个强连通分量的权值为一个二元组，为这个强连通分量中的最大值和次大值（如果次大值没有置为空即可）。然后我们考虑在缩点后的 DAG 上递推标记，每个点的标记由其入边的对应的点更新即可。

实现的时候可以枚举一个位置，只需知道能到这个位置的最大值就行了

51NOD 路径定向

- 给出一个有向图，要求给每条边重定向，使得定向后出度等于入度的点最多，输出答案和任意一种方案

- Input
- 第一行两个正整数N，M，表示1-N号点与M条边
- 接下来M行，每行两个正整数Xi，Yi，表示存在一条有向边从Xi指向Yi
- N≤10^5，M≤3*10^5，Xi,Yi≤N

- 尝试进行构造
- 由出度和入度相等想到欧拉回路
- 考虑对任意的方案，如果在奇点间随意连边，并不影响结果
- 所以我们可以在起始时在奇点间连辅助边，获得一张只有偶点的图
- 然后跑一遍欧拉回路定向即可

- 答案为偶点的个数

BZOJ 4484 [JSOI2015]最小表示

- 对于一个N个点（每个点从1到N编号），M条边的有向图，JYY发现，如果从图中删去一些边，那么原图的连通性会发生改变；而也有一些边，删去之后图的连通性并不会发生改变。
  JYY想知道，如果想要使得原图任意两点的连通性保持不变，我们最多能删掉多少条边呢？
  为了简化一下大家的工作量，这次JYY保证他给定的有向图一定是一个有向无环图。

- N<=30,000,M<=100,000

- 如果从u到v有(u,v)这条边,且从u到v只有这一条路径,那么这条边必须保留.否则这条边一定可以删除.因为如果有不止一条路径从u到v,必然存在点x(x!=u,x!=v)使得u可到达x,x可到达v.而删边后必然也满足u可到达x,x可到达v,所以直接删掉(u,v)这条边就可以了.
  刚才的分析已经给出了一个判定方法.既然如果有不止一条路径从u到v,必然存在点x(x!=u,x!=v)使得u可到达x,x可到达v,那么我们对每条边(u,v),枚举是否存在这样的x即可.这需要我们求出每个点能到达的点的集合,以及能到达这个点的集合.大力压位一波就好了.因为是DAG所以这个集合可以递推.
  复杂度O(nm/32).