Day1 题解

成都市第七中学 袁方舟

2019 年 5 月 29 日

选w次，每次随机选择一个三维长方体，问不经过障碍的同时至少被选过一次的点的期望权值和。

$n, m, k \leq 60, w \leq 10^9$

梦批紡

得分分布

考虑容斥，对于一个点算不经过它的概率。那么只需要算不经过这个点且不经过障碍的长方体个数。

考虑容斥，对于一个点算不经过它的概率。那么只需要算不经过这个点且不经过障碍的长方体个数。
超级暴力：$O(n^9)$
枚举每个长方体然后枚举这个长方体里的每个点，期望得分:10

考虑容斥，对于一个点算不经过它的概率。那么只需要算不经过这个点且不经过障碍的长方体个数。
超级暴力：$O(n^9)$
枚举每个长方体然后枚举这个长方体里的每个点，期望得分:10
普通暴力：$O(n^6)$
稍微用前缀和优化一下超级暴力，期望得分:40

记dp[8][x][y][z]表示8个方向（$\leq x, \leq y, \leq z; \leq x, \leq y, \geq z; \leq x, \geq y, \leq z; \leq x, \leq y, \geq z; \geq x, \leq y, \leq z; \geq x, \leq y, \geq z; \geq x, \geq y, \leq z; \geq x, \leq y, \geq z;$）里包含的长方体个数。

记dp[8][x][y][z]表示8个方向（$\le x, \le y, \le z; \le x, \le y, \ge z; \le x, \ge y, \le z; \le x, \le y, \ge z; \ge x, \le y, \le z; \ge x, \le y, \ge z; \ge x, \ge y, \le z; \ge x, \le y, \ge z;$）里包含的长方体个数。
这个dp可以枚举每层然后用单调栈算出。

记dp[8][x][y][z]表示8个方向（$\leq x, \leq y, \leq z; \leq x, \leq y, \geq z; \leq x, \geq y, \leq z; \leq x, \leq y, \geq z; \geq x, \leq y, \leq z; \geq x, \leq y, \geq z; \geq x, \geq y, \leq z; \geq x, \leq y, \geq z;$）里包含的长方体个数。
这个dp可以枚举每层然后用单调栈算出。
那么不包含点(x, y, z)的长方体个数可以由这个dp得到。

记dp[8][x][y][z]表示8个方向（$\leq x, \leq y, \leq z; \leq x, \leq y, \geq z; \leq x, \geq y, \leq z; \leq x, \leq y, \geq z; \geq x, \leq y, \leq z; \geq x, \leq y, \geq z; \geq x, \geq y, \leq z; \geq x, \leq y, \geq z;$）里包含的长方体个数。
这个dp可以枚举每层然后用单调栈算出。
那么不包含点(x, y, z)的长方体个数可以由这个dp得到。

时间复杂度$O(n^4)$，期望得分:100

等你哈苏德

题意

数轴上有许多线段，你需要将他们染成黑色或白色，使得对于数轴上每个整点，覆盖它的黑色线段和白色线段个数之差绝对值不超过1。

$m \leq 3 \times 10^4, n \leq 10^9$

首先考虑所有线段都没染色的情况。

首先考虑所有线段都没染色的情况。

离散化后，我们只需要关心每个端点处的差值。如果把区间$[l, r]$视为l和r之间连了一条无向边，染色视作是定向，我们可以发现这就是一个欧拉路径问题。我们把奇度数的点排序后依次连接，跑欧拉回路即可。

首先考虑所有线段都没染色的情况。

离散化后，我们只需要关心每个端点处的差值。如果把区间$[l,r]$视为l和r之间连了一条无向边，染色视作是定向，我们可以发现这就是一个欧拉路径问题。我们把奇度数的点排序后依次连接，跑欧拉回路即可。

一部分线段染色相当于是混合图欧拉回路，跑网络流即可。由于这是个单位流量的图，所以dinic算法可以快速跑出。

一棵有边权的树，要求用最小的代价遍历这棵树。对于所有的K，计算允许加入不超过K条额外边后遍历树边的最小代价。
$n, K \leq 10^5$

喜欢最最痛

得分分布

观察到这个题的答案实际上分成两部分：对于一个K，$dp[i]$表示加i条额外边（它们的边权为0）且必须走的代价，和前K个边权中的前i小之和。

观察到这个题的答案实际上分成两部分：对于一个K，$dp[i]$表示加i条额外边（它们的边权为0）且必须走的代价，和前K个边权中的前i小之和。

又可以观察到加i条额外边最小边权和就是在这个树上选i条边不相交的链使得权值最大。

观察到这个题的答案实际上分成两部分：对于一个K，$dp[i]$表示加i条额外边（它们的边权为0）且必须走的代价，和前K个边权中的前i小之和。

又可以观察到加i条额外边最小边权和就是在这个树上选i条边不相交的链使得权值最大。
使用树形Dp即可，复杂度$O(n^2)$

可以很容易知道（猜）答案是一个凸包。但是现在我们需要求出所有$O(n)$个答案。

可以很容易知道（猜）答案是一个凸包。但是现在我们需要求出所有$O(n)$个答案。
考虑使用与费用流类似的“悔过”操作，每次在这棵树上找直径，然后将直径的边翻转成负的。因为有负边权，std使用LCT维护动态DP实现，常数稍大。

可以很容易知道（猜）答案是一个凸包。但是现在我们需要求出所有$O(n)$个答案。
考虑使用与费用流类似的“悔过”操作，每次在这棵树上找直径，然后将直径的边翻转成负的。因为有负边权，std使用LCT维护动态DP实现，常数稍大。
最后就是算最终答案的问题了。由于两部分答案都是凸的，所以直接二分就行了。时间复杂度$O(n\log n)$或$O(n\log^2 n)$均可通过该题。

杂题选讲I

成都市第七中学 袁方舟

2019 年 5 月 29 日

XorTable

有两个数组$A_1, \ldots, A_n$和$B_1, \ldots, B_m$。你不知道它们的值，你只知道$A_i \in [L_i, R_i], B_i \in [P_i, Q_i]$。你还知道一个$n \times m$的矩阵$C$，其中$C_{i,j} = A_i \oplus B_j$或者$C_{i,j} = -1$表示未定义。给出这个矩阵$C$以及限制，要求任意一组解或者判定无解。
数据范围：$n, m \le 10^3, C_{i,j}, L_i, R_i, P_i, Q_i \le 10^9$

首先把这个矩阵有值的点当做边，这样可以对于每个连通块求解。现在问题转化为给出若干个L_i, R_i, x_i，要求对$[L_i, R_i] \oplus x_i$这些集合进行求交。

首先把这个矩阵有值的点当做边，这样可以对于每个连通块求解。现在问题转化为给出若干个L_i, R_i, x_i，要求对$[L_i, R_i] \oplus x_i$这些集合进行求交。可以将这个区间分成$O(\log)$长为2^k的区间，这个区间异或x后仍然是一个区间。对这些区间求交就可以了。时间复杂度$O((n+m)\log)$。

Yandex.Algorithm 2018 final round

LIS vs. LDS

给定一个长为n的排列，你需要找一个和LIS长度相等的子序列以及一个和LDS长度相等的子序列，使得这两个子序列之间没有交点。

$n \leq 500000$

LIS vs. LDS

题解

考虑如何判定无解。我们尝试计算出所有的方案数和只选取相交的子序列的方案数。然后比较是否相等。

LIS vs. LDS

题解

考虑如何判定无解。我们尝试计算出所有的方案数和只选取相交的子序列的方案数。然后比较是否相等。
先计算出全局的LDS数量和经过每个数的LDS数量。然后DP计算带权LIS，每个数的权就是经过每个数的LDS数量，这是因为LIS和LDS最多一个交点。

考虑如何判定无解。我们尝试计算出所有的方案数和只选取相交的子序列的方案数。然后比较是否相等。
先计算出全局的LDS数量和经过每个数的LDS数量。然后DP计算带权LIS，每个数的权就是经过每个数的LDS数量，这是因为LIS和LDS最多一个交点。
对随机模数取模就可以保证错误率很低。时间复杂度$O(n \log n)$。

LIS vs. LDS

题解

考虑如何判定无解。我们尝试计算出所有的方案数和只选取相交的子序列的方案数。然后比较是否相等。
先计算出全局的LDS数量和经过每个数的LDS数量。然后DP计算带权LIS，每个数的权就是经过每个数的LDS数量，这是因为LIS和LDS最多一个交点。
对随机模数取模就可以保证错误率很低。时间复杂度$O(n \log n)$。
输出方案的话就先找一个合法的LIS，然后再在剩余序列中找一个LDS即可。

Codeforces 627F

橡树的一部分是一棵n个点的树，每个点上有一片树叶，这些树叶两两不同，记第i个点上的树叶为ai.有一个点上没有树叶，这个点的ai=0.保证ai是一个[0...n-1]的排列。
你手上有个排列$\{b_i\}$，橡树上的排列为$\{a_i\}$，如果一条边的一端树叶为空，你可以选择把这条边一端的树叶移到另一边。你坚信将$\{a_i\}$转换为$\{b_i\}$就可以摧毁橡树。同时，你发现在必要的时候你可以加一条边（把树的一部分变成环）。你想要知道最少的操作次数，和这个边加在哪或者不加。

$n \leq 10^5$

首先思考如何判定有解。

首先思考如何判定有解。

首先，可以先把0移到指定位置。由于可以逆操作，所以这不影响有解性。

首先思考如何判定有解。

首先，可以先把0移到指定位置。由于可以逆操作，所以这不影响有解性。
如果所有点满足条件，那么不用加边，直接输出。

首先思考如何判定有解。

首先，可以先把0移到指定位置。由于可以逆操作，所以这不影响有解性。
如果所有点满足条件，那么不用加边，直接输出。
令0的位置为根，现在树上$a_i \neq b_i$的点有两种情况：
(1)这些点构成了一条祖先到后代的链。
(2)这些点构成了一条不是祖先到后代，且lca处满足$a_i = b_i$的链。
同时，这些点也必须构成环。这唯一确定了加的环边。

现在考虑如何算最少步数。

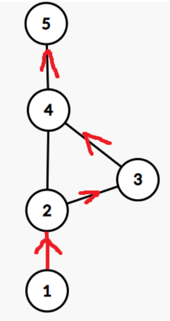
现在考虑如何算最少步数。
如果移动0的时候没有经过这个环，那么就必须花额外的代价走到这个环，然后再走回去。这时答案就是两种转换方法取min

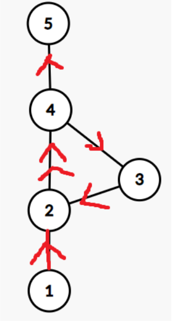
CodeForces 627F

计算答案

现在考虑如何算最少步数。
如果移动0的时候没有经过这个环，那么就必须花额外的代价走到这个环，然后再走回去。这时答案就是两种转换方法取min
如果经过了这个环，可以把这个环搞好之后再移动。

现在考虑如何算最少步数。
如果移动0的时候没有经过这个环，那么就必须花额外的代价走到这个环，然后再走回去。这时答案就是两种转换方法取min
如果经过了这个环，可以把这个环搞好之后再移动。





CodeForces 611H

看不见的树

神树大人手上有一个n个点的树。这个树由$n-1$条边(u_i, v_i)构成。可惜的是，这些数对都被遮住了，神树大人现在只知道u_i和v_i在十进制下的位数。神树大人想要还原原来的树，或者判定无解。
比如，(1,3)(2,10)被遮住之后就是(?,?)(?,??)这样。
$n \leq 2 \times 10^5$

节点可以按数位个数被分成$k = 6$组。

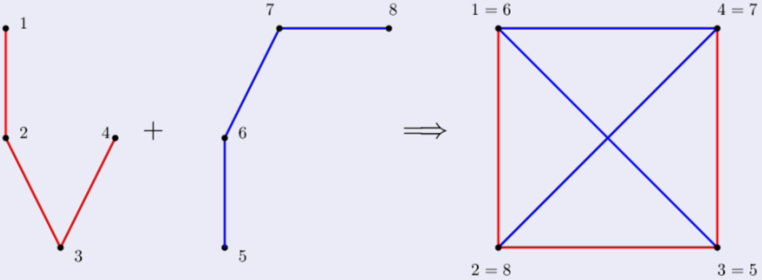
节点可以按数位个数被分成$k = 6$组。
易证有一种方案如下：可以在每一组都选择一个“关键点”，其他的点都只和这些关键点相连。

节点可以按数位个数被分成$k=6$组。
易证有一种方案如下：可以在每一组都选择一个“关键点”，其他的点都只和这些关键点相连。
枚举树的形状k^{k-2}，跑网络流即可。

CodeForces 927F

Public Service

给两棵树，求一个排列P使得不存在(i,j)和(p_i,p_j)同时在两棵树中分别有边。



$n \leq 10000$

菊花图显然无解。可以证明其他情况必定有解。

菊花图显然无解。可以证明其他情况必定有解。
(1)n很小的时候$O(n!)$求解。

Public Service
题解

菊花图显然无解。可以证明其他情况必定有解。
(1)n很小的时候$O(n!)$求解。
(2)假设有一棵树删一个点就变成菊花图。
那么找到那个点 v 以及跟它相连的 u 和跟 u 相连的 w 。
在第二棵树中，找到一个叶子 $w2$ 和跟它相连的 $v2$ 。并暴力寻找跟 $v2$ 和 $w2$ 都不相连的点 $u2$ ，由于第二棵树不是菊花图，所以 $u2$ 一定存在。

菊花图显然无解。可以证明其他情况必定有解。
(1)n很小的时候$O(n!)$求解。
(2)假设有一棵树删一个点就变成菊花图。
那么找到那个点 v 以及跟它相连的 u 和跟 u 相连的 w 。
在第二棵树中，找到一个叶子 $w2$ 和跟它相连的 $v2$ 。并暴力寻找跟 $v2$ 和 $w2$ 都不相连的点 $u2$ ，由于第二棵树不是菊花图，所以 $u2$ 一定存在。
(3)其他情况则各找一对叶子节点对应。

投票选举

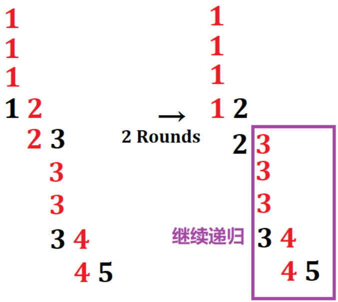
m个人竞选最神的人，有n个人投票。一开始每个人心里都有个m个人中一些人的按神神度排序之后的排列（不一定要出现所有人），接下来做如下操作：
(1)第一轮时每个人投票给列表第一个人
(2)每个人把自己的投票选项改成自己列表中得票最多的那个，如果有多个则选最靠前的
(3)如果每个人的选项都没变则结束，否则重新投票并跳到(2)
现在请对每个人都构造一个排列，使得投票的轮数尽量多。要求列表总大小不得超过20000，$n \leq 10000, m \leq 20000$，轮数至少为4999。

我们考虑构造一堆“形式相同”的投票轮。

投票选举

一种构造方法

我们考虑构造一堆“形式相同”的投票轮。



其中红色代表每个人的选项。这样可以构造出0.5n左右的答案，如有更优秀的方法（比如指数级构造或者只是大于该复杂度的构造）可以讨论。

普及组题

有一棵n个点的树，我把原树复制了一棵，然后在这两棵树里面各删去一个点（不能删去编号相同的点），然后打乱点的编号。现在给出这两个删点后的森林，你需要复原出原树。

$n \leq 1000$

两个森林枚举再删一个点，然后算树哈希。

两个森林枚举再删一个点，然后算树哈希。
时间复杂度$O(n^2\log n)$，如有更好的算哈希方法可以$O(n^2)$。

Algoland and Berland

平面上有a个红色点和b个蓝色点，没有三点共线，红蓝点之间可以连边，求一棵生成树，使得第i个蓝点的度数恰好为r_i。
$a, b \leq 3000$，保证$\sum r_i = a + b - 1, 1 \leq r_i \leq a$。

考虑递归构造，首先$\max r_i = 1$的时候构造显然：选一个红色点连就行了

考虑递归构造，首先$\max r_i = 1$的时候构造显然：选一个红色点连就行了
否则取度数最大的点，用一条直线将这个平面划成两部分递归做。

考虑递归构造，首先$\max r_i = 1$的时候构造显然：选一个红色点连就行了
否则取度数最大的点，用一条直线将这个平面划成两部分递归做。
记一个点的权值为$r_i - 1$（红点的权值为0），我们需要将这个平面划成两部分使得两部分的权值和≤ -1，这样我们就可以保证一定有解：因为更小的情况已经被归纳证明了。

考虑递归构造，首先$\max r_i = 1$的时候构造显然：选一个红色点连就行了
否则取度数最大的点，用一条直线将这个平面划成两部分递归做。
记一个点的权值为$r_i - 1$（红点的权值为0），我们需要将这个平面划成两部分使得两部分的权值和≤ -1，这样我们就可以保证一定有解：因为更小的情况已经被归纳证明了。
令一个半平面的和为sum，那么我们需要找一个$1 - \max r_i \le sum \le -1$。
很容易证明可以找到。

候选队互测

绝目编诗

给出一张简单无向图，判断是否存在两个长度相同的简单环。

$n \leq 5000$

先考虑一个$O(n^2m)$的算法。

先考虑一个$O(n^2m)$的算法。
枚举每个点，找出包含它的所有环，然后删除这个点。

先考虑一个$O(n^2m)$的算法。
枚举每个点，找出包含它的所有环，然后删除这个点。
每次枚举这个环的下一步，并判连通性。

我们猜测m很大的时候答案一定是yes。

我们猜测m很大的时候答案一定是yes。
当$m > 2n$的时候，根据抽屉原理可以证明答案一定是yes。所以上述算法复杂度降为$O(n^3)$。

我们猜测m很大的时候答案一定是yes。
当$m > 2n$的时候，根据抽屉原理可以证明答案一定是yes。所以上述算法复杂度降为$O(n^3)$。
实际上可以证明$m - n > O(\sqrt{n})$时答案一定是yes。

每次删去一个最多环覆盖的边

每次删去一个最多环覆盖的边

设$l = m - n + 1$，$n \geq A_0 > A_1 \ldots > A_l = 0$为删去$i$条边之后环的个数。

每次删去一个最多环覆盖的边

设$l = m - n + 1$，$n \geq A_0 > A_1 \ldots > A_l = 0$为删去$i$条边之后环的个数。

有$A_i - A_{i-1} \geq \max(C \times \frac{A_i^2}{n}, 1)$（$C$是常数），考虑环的总长度至少为$\frac{1}{2}A_i(A_i + 1)$，由抽屉原理可得。

每次删去一个最多环覆盖的边
设$l = m - n + 1$，$n \geq A_0 > A_1 \ldots > A_l = 0$为删去$i$条边之后环的个数。
有$A_i - A_{i-1} \geq \max(C \times \frac{A_i^2}{n}, 1)$（$C$是常数），考虑环的总长度至少为$\frac{1}{2}A_i(A_i + 1)$，由抽屉原理可得。
可以得到$l \leq O(\sqrt{n} + \sum_{i=1}^{\sqrt{n}} \frac{\sqrt{n}}{i^2}) = O(\sqrt{n})$，考虑每个$[i\sqrt{n}, (i+1)\sqrt{n})$的表现即可（这个证明无比丑陋）

每次删去一个最多环覆盖的边
设$l = m - n + 1$，$n \geq A_0 > A_1 \ldots > A_l = 0$为删去$i$条边之后环的个数。
有$A_i - A_{i-1} \geq \max(C \times \frac{A_i^2}{n}, 1)$（$C$是常数），考虑环的总长度至少为$\frac{1}{2}A_i(A_i + 1)$，由抽屉原理可得。
可以得到$l \leq O(\sqrt{n} + \sum_{i=1}^{\sqrt{n}} \frac{\sqrt{n}}{i^2}) = O(\sqrt{n})$，考虑每个$[i\sqrt{n}, (i+1)\sqrt{n})$的表现即可（这个证明无比丑陋）
所以使用虚树运行算法1即可，复杂度$O((\sqrt{n})^3) = O(n\sqrt{n})$。

Ending

感谢大家的聆听!