高子翼集训队试题试题准备

Tavas in Kansas

试题来源

Codeforces Round #299 (Div. 1) D

试题大意

给定一张n个点,m条边带边权的无向图G，每个点都有权值p_i。A,B二人在G上进行一个游戏。

开始时A在s点，B在t点，他们需要轮流操作，每次操作者选择一个x，并获得所有离它距离不超过x的点的权值。获得完某个点的权值之后，这个点的权值将变为0并打上标记。每次操作，必须获取至少一个未被标记过的点的权值。

两人轮流操作，都将使用最优策略，A先手，请问是否最后A获得的权值比B多。

数据范围

$2 \le n \le 2000$

$n-1 \le m \le 10^5$

保证图联通

可能存在重边自环。

解题报告

首先跑出s，t分别为起点的最短路，图上每个点我们给他一个坐标分别为(x_i, y_i)，分别代表其到s，t的距离。

那么博弈过程就是每次取横坐标或纵坐标小于等于某值的点。

由于是博弈问题，所以需要从最终状态转移回开始状态。

将坐标离散化，记$dp_{i,j,0/1}$代表取了$x_i > i$和$y_i > j$的先手与后手最大权值之差，0/1代表A是先手还是B是先手。

转移十分显然，枚举上一回合的先手的策略，可以使用简单的后缀max优化成$O(n^2)$。

Grafting

试题来源

AtCoder Grand Contest 027 F

题目大意

给定两棵n个节点的树A,B，你需要执行若干次操作，每次操作选择一个A的叶子v，删掉与其相邻的边，并加入一条边连接v和另一个点，每个点只能被选择一次。请操作最少的步数使A,B相同。

数据范围

$1 \leq T \leq 20$

$1 \leq n \leq 50$

解题报告

首先假设某个点未被操作过，那么我们把这个点当做根。

我们分别dfs得到两棵树的父亲数组$father_A[i]$和$father_B[i]$，两棵树同构当且仅当$father$数组相同。

可以轻松得到一个性质，如果$t = father_A[i] = father_B[i]$，那么我们肯定不会操作$i$这个点，同时也就不会操作$t$，所以$father_A[t] = father_B[t]$，以此类推，$i$在$A$树上到根的链必定和$B$树上到根的链相同，这个可以简单dfs判断。

不难发现，$father_A[i] \neq father_B[i]$的点是必须被操作的，所以操作次数和操作结果就固定了，考虑找到一个操作的顺序即可。

对于A树上的点i，i的操作时间必须要比$father_A[i]$早，对于B树上的点i，i的操作时间必须要比$father_B[i]$晚。并且如果操作满足这个顺序，那么一定合法，因为当操作i时，在A的以i为根的子树中一定都已经操作完(一定为叶子)，在B中的父亲也已经操作完。那么我们只需要判断这种先后关系是否出现环即可。

Sequence Growing Hard

题目来源

AtCoder Grand Contest 024 E

题目大意

请求出可能的以序列为元素的，满足以下条件的多元组$(A_0, A_1, A_2 \ldots\ldots, A_n)$的组数，并且对$m$取模。

- 对于任意$0 \leq i \leq n$，A_i的长度为i，并且其包含的元素均为1到k中的整数。
- 对于任意$1 \leq i \leq n$，A_{i-1}是A_i的子序列，也就是说，存在x_i使得移除A_i中的第x_i个元素之后，序列A_i与序列A_{i-1}相同。
- 对于任意$1 \leq i \leq n$，A_i的字典序严格大于A_{i-1}。

数据范围

$1 \le n, k \le 300$

$2 \le m \le 10^9$

题解报告

首先我们给每个A_i的末尾添上一个0。

考虑从A_i到A_{i+1}只增加了一个数，不妨设为x,假设我们将x添加在y之前，那么需要满足以下任意一个条件使得其合法：

1. $x > y$。
2. $x = y$,并且在y之后的第一个非y元素小于x。

由于对于同一A_i来说，方案的不同是由A_{i+1}是否相同来决定的，而不是由添加的位置和元素大小来决定的，所以上面这种添加方式得到的A_{i+1}可能是有重复的（当然，不会遗漏）。

只考虑第1种添加方式，显然根据这种方式得到的A_{i+1}不会重复。

然而对于第2种添加方式而言，得到的A_{i+1}必定都可以通过第1种添加方式得到。因为考虑A_i的一段中有$yy \cdots yt(y > t)$这样的序列，然后我们在第一个y前加上一个y，如果我们在t前加上一个y，得到的序列是一样的。

那么我们只需要考虑第一种添加方式即可。

我们考虑把一种方案对应到一棵树上去，考虑构造一棵$n+1$个点的有根树，每个点上面有一个二元组(x_i, id_i)。$x_i \in [0, k], id_i \in [0, n]$，并且$id_i$互不相同

若保证某个点为i，其父亲为t，那么有$x_i > x_t, id_i > id_t$。树的根为$(0, 0)$。

树不同当且仅当二元组不同或者父亲不同，不难发现不同的树个数就是序列的方案数。（id为其插入的时间，x代表插入的值）

考虑统计这种树的个数。

考虑$dp[i][j]$代表i个点的树，根的值为j的方案数。转移可以枚举$id=1$的那棵子树来实现。

$dp[i][j] = \sum_{a=1}^{i-1} \sum_{l=j+1}^{k} dp[i-a][j] \times dp[a][k] \times \binom{i-2}{a-1}$

答案就是$dp[n+1][0]$

使用简单的前缀和技巧可以做到$O(n^2 k)$，这个做法可以通过原题。

当然我们可以进一步优化。

考虑设$F_j(x) = \sum_{i=1}^{\infty} \frac{dp[i][j]}{(i)!} x^i$,$G_j(x) = \sum_{l=j}^{k} F_l(x)$。

不难发现，一棵树可以由一些权值大于它的根的权值的子树构成，并且这些子树是无序的，所以我们可以得到$F_j(x) = \int e^{G_{j+1}(x)} \mathrm{d}x$。(根是$id$最小的，其他的可以自由排列）

通过多项式exp技巧可以将复杂度优化至$O(nk \log n)$

不难发现答案是关于k的n次多项式（考虑选出树的权值之后离散化），那么我们只需要算出k为0~n的答案，然后插值即可，可以将复杂度做到$O(n^2 \log n)$。

上面的式子貌似还有进一步优化的空间，如果能得到更优的做法欢迎与本人交流。