Day3 讲题 + 杂题选讲

Scape

June 11, 2018

得分情况

100 分:22 人
为什么会这样呢。

先来分析点性质

只考虑 $L \geq 4$ 的情况。
? 是一个长度为 L-3 的块
a?bcd-$>$ca?bd-$>$cda?b-$>$ca?bd-$>$bca?d-$>$a?dbc
也就是说任意长度为 3 的都可以被移位。

先来分析点性质

a?bcd -> a?cdb -> ab?cd
也就是说任意长度为 $L-2$ 的都可以移位。

分情况讨论

如果 L 是偶数，那么显然任意相邻的都可以换。
$\binom{N+K-1}{K-1}$

分情况讨论

如果 L 是奇数。

首先如果有两种相同的颜色，那么还是任意相邻的可以换。

?a?a?bc?->aabc????->acab????->aacb????->?a?a?cb?

颜色互不相同，则有

?a?b?->ab

ab?cd?->abcd??->badc??->ca?dc ?

所以有 2 种。

$\binom{N+K-1}{K-1} + \binom{K}{N}$

得分情况

为什么会这样呢。

题解

什么情况会 0：四角上有 x，或者第一行和第三行连续 2 个 x。
注意到联通块之间互不影响，分别算之后合并就可以了。
从左往右 dp,$f(i,j)$ 表示在点 i，点 i 的顺序是第 j 个, 然后点 i 是竖着来的方案数。
$g(i,j)$ 表示在点 i，点 i 的顺序是第 j 个, 然后点 i 是横着来的方案数。
转移就讨论 $i+1$ 的情况就可以了。
为了保证不重复数要保证 $g(i,j)$ 的时候竖着的不能被填满（不然的话会在 $f(i,j)$ 里也算一次)。

得分情况

为什么会这样呢。

题解

暴力？
把依赖关系建出来每次看后继的 size
$O((rc)^2)$

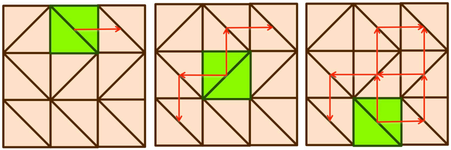
题解

只要从四方优化成三方，看起来不用太难的优化？
同一列的依赖关系是单调的。一起做
$O(r^2c)$

题解

只要从四方优化成三方，看起来不用太难的优化？
同一列的依赖关系是单调的。一起做
$O(r^2c)$

题解



Path

给定 n 和 a_i, 满足 $a_0 \geq a_1 \geq \cdots a_{n-1} \geq 0$, 求出在 n 维空间中从 $(0, 0, \cdots, 0)$ 走到 $(a_0, a_1, \cdots, a_{n-1})$, 每一步使某一维坐标增加 1 的方案中随机选出一种, 满足经过的所有点 $(x_0, x_1, \cdots, x_{n-1})$ 都满足 $x_0 \geq x_1 \geq \cdots \geq x_{n-1}$ 的概率。答案模 1004535809 输出.
$n, a_i \leq 10^6$

Path

我们发现, 这道题实际上就是要求将 $1 \sim n$ 这 n 个数字填入这个图中, 使得每行从左到右都是递增的，每列从下到上也是递增的。考虑用 Hook length formula。就是要算

$$\prod_{i,j} h_{i,j} = \prod_{i} \frac{(a_i + n - i - 1)!}{\prod_{j \geq i}(a_i - a_j + j - i)}$$

$$\prod_{i} (a_i + n - i - 1)! \prod_{i \leq j} \frac{1}{\prod_{j \geq i}(a_i - i) - (a_j - j)}$$

FFT 一下。

简单数据结构题

给一棵 n 个点的树，点权开始为 0，有 q 次操作，每次操作是选择一个点，把周围一圈点点权 $+1$（一个点周围的点为与该点距离为 1 的点），在该操作后你需要输出当前周围一圈点点权的异或和。

$n, q \leq 5 \times 10^5$

简单数据结构题

我们把每个节点的儿子集合搞出来。
然后每次就等于是一个集合 $+1$ 或者单点 $+1$。
然后异或和可以按位拆开。
等于对于位 i，我们要维护每个点 $\mathrm{mod} 2^{i+1}$ 的值，当它 $\geq 2^i$ 的时候这一位就是 1。
注意到只有集合整体加 1 和单点加 1 的时候这也是很好线性维护的。(直接开个 $cnt[i]$ 表示 i 有多少个就可以了)。

Dagon

给定 n 个区间 l_i, r_i，给每个区间染上 k 种颜色之一，使得每种颜色的区间的交长度之和最大。
$n, k \leq 10^6$

Dagon

如果存在答案区间为空的，那么最优的肯定是前 $k-1$ 大的。
注意如果一个区间覆盖另外一个区间，那么在考虑并的时候是可以不考虑它的，可以暂时去除。
接下来只要考虑不存在答案区间为空的，也就是答案区间一定是某个 $\min\{r_i\}-\max\{l_i\}$。
注意到合并两端相邻的区间 $[l_i,r_i]$,$[l_{i+1},r_{i+1}]$，减少的答案为 $r_{i+1}-l_i$，新区间为 $[l_{i+1},r_i]$。
这说明合并答案减少量是固定的，不随区间改变而合并。
预处理减少量，排序之后贪心计算。每次用当前代价 + 去除的区间中前剩下可用的颜色数大的区间长度和来更新答案。

An unavoidable detour for home

求满足下列条件的 n 个点的无向图数量:
- 它没有重边自环，所有边的权均为 1
- 从 1 号点到任意点的最短路都是唯一的. 设 l_i 是从 1 号点出发到达 i 号点的最短路长度，你虽然不知道 l_i, 但知道 l_i 关于 i 单调不减。
- 你知道第 i 个点连了 d_i 条边, 并发现它们都要么为 2 要么为 3。

由于答案可能很大，模 $10^9 + 7$ 输出
$n \leq 50$

An unavoidable detour for home

按照距 1 号点的距离分层，分层，发现每层在 1 到 n 的序列上是一个区间，并且后一层的距离等于前一层的距离 $+1$ 。除 1 号点外，每个点应当恰好向前一层连 1 条边 (为了确保最短路唯一）并且向当前层和下一层任意连边。
$f(i, pre1, pre2, now1, now2)$ 表示确定了前 i 个点间互相连的边，当前层（即 i 号点所在的层）有 $now1$ 个点剩余 1 条边需要连，$now2$ 个点剩余 2 度数，前一层有 $pre1$ 个点剩余 1 度数，$pre2$ 个点剩余 2 度数时的方案数。
转移的时候直接讨论转移就可以了。

Games on DAG

有一张 N 个点 M 条边的有向图 G 。
考虑选取一个图 G 的 M 条边的子集，移除这些边来生产一个新的图 G 。显然图 G 有 2^M 个。
Alice 和 Bob 正在图 G 上玩一个游戏，初始，在节点 1 和 2 上面各有一个棋子。接着他们开始玩 Nim 游戏。
在所有不同的 2^M 个图 G 里面有多少能够使得 Alice 获胜？请求出答案对 10^9+7 取模后的值。
$n \leq 15$

Games on DAG

注意到按照 SG 值有个天然的分层。
按照 SG 值分层，第 0 层没有出边，第 i 层的每个点会向 $< i$ 层的每一个层至少一个点连边。
令 f_S 表示考虑了点集 S 的删边方案。
枚举新的一层点集 S_1 转移就可以了。
S_1 内部的边肯定要全部删。
然后如果指向 SG 值更小的边，就是每一层至少要保留一条。往更大的就随便了。

Yes or No

目前有 $N+M$ 个问题摆在你面前。
每个问题的答案是 Yes 或 No，你知道答案为 Yes 的问题恰好有 N 个，答案为 No 的问题恰好有 M 个，但是问题是随机给出的，你完全不知道每个问题的答案分别是多少。
你现在可以逐个去猜每个问题的答案，每猜完一个问题的答案，你就会立刻知道这个问题真正的答案。
如果你用最优的策略去猜各个问题的答案，请求出你猜对的次数的最大的期望值。
对 998244353 取模。
$n, m \leq 500000$

Yes or No

不妨设 $n \geq m$
最优策略，选择剩余多的，一样多乱猜一个。
你假设从一个 (i,i) 走到 $(0,0)$ 中途不到对角线，那么显然你会一直猜同一个，一定会答对 i 个。
发现从 (n,m) 到 $(0,0)$ 无论中途多曲折，经过对角线多少次，我们一个部分一个部分分开，都会答对对应次。
因此无论如何都会答对 n 次。如果走到对角线上，我们会乱猜，只有 $\frac{1}{2}$ 几率对。
因此对于对角线上每一个点统计经过它的方案数即可。预处理组合数就能直接算了。

欧拉子图

给定一个 n 个点，m 条边无向图（可能有重边），对于这个图的边集的子集（一共有 2^m 个），如果其导出的子图的每个联通块内都存在欧拉回路，我们就把答案加上这个子图的边数的平方，答案对 10^9+7 取模。$n, m \leq 200000$

欧拉子图

对每两条不同的边 a b, 计算 $E(a,b)$ 表示 a 和 b 都选了的合法子图有几个。对每条边 a，计算 $E(a)$ 表示选了 a 的合法子图有几个。容易看出答案是所有 $E(a,b)$ 的和乘 2 加上所有 $E(a)$ 的和。这样我们就可以通过枚举 a b 之后统计包含枚举的边的方案数。我们可以用线性基的经典做法来求解。即有 m 个 01 变量（代表每条边是否选），要满足 n 个异或方程（每个点的度数是偶数），并且要求其中两个变量为 1（a、b 对应的变量）的方案数。

欧拉子图

观察到不同连通块间一定线性无关，而一个连通块的秩为 $V-1$。
由此可得，对于每个连通块，设连通块点数 V，边数 E，则该连通块自由元的数量是 $E-V+1$，
欧拉子图方案数是 2^{E-V+1}。
设连通块数量为 s，枚举 a b，如果 a b 中有桥边，则一定无解。
如果删去 a b 后连通性不变，则等价于将 a b 连通块自由元的数量各减 1（因为 x_a x_b 被限制等于 1），方案数 $2^{n-m+s-2}$。如果删去 a b 后连通块多了一个，则主元与自由元各少了一个，方案数 $2^{n-m+s-1}$。

欧拉子图

所以只需求有多少组 a, b 使得删去 a, b 后连通性改变。
给每条非树边随机一个 unsigned long long 的权值，每条树边的权值等于所有跨过它的非树边的权值的异或和，如果 a, b 权值相同，连通性就改变。
证明: 假设删掉边集后整个图分成了两个互不连通的点集。考虑边集中所有跨过了两个点集的边（包括树边和非树边）的异或和。对于一条两端都在同侧的非树边，它的权值显然会被树边计算偶数次；而对于两端在异侧的非树边，它的权值会被树边计算奇数次，再加上它本身算的一次后总共也被算了偶数次。因此这些边的异或和必定为 0。

Many Easy Problems

给定一棵 N 个节点的树与一个整数 K.
对于一个顶点集合 S, 定义 $f(S)$ 为最小的包含 S 中所有点的连通块大小。
有 $\binom{N}{K}$ 种方案从 N 个点中选取一个大小为 K 的点集，设被选择的点集为 S, 求出 $\binom{N}{K}$ 种方案中 $f(S)$ 的和。答案可能很大，只需求出它模 924844033 的值。
因为这对他来说太简单了，他决定对于 $K = 1, 2, ...N$ 的每一种情况求出问题的解。
$n \leq 200000$

Many Easy Problems

考虑某条边对答案的贡献。假设它把树分成了一个大小为 S 和一个大小为 $N-S$ 的子树。如果两个子树中都有被选择的点，则这条边会对答案贡献 1。

$$\binom{N}{K}-\binom{S}{K}-\binom{N-S}{K}$$

可以 NTT。

count

有 N 个点, 每个点有一个权值 A_i. 现在要把这 N 个点连成一棵树, 设第 i 个点在树中的度数为 d_i, 那么这棵树的权值为 $\prod d_i A_i^{d_i}$。
求所有可能的树的权值之和模 10^9+7 的结果。
$N \le 2000; Ai \le 1000000.$

count

考虑 prufer 序列。
答案显然是
dp 一下是 n^3 的，考虑优化。

count

$$\sum_{d_1+d_2+\ldots+=2n-2} \frac{(N-2)!}{(d_1-1)!(d_2-1)!\cdots} \prod d_i A_i^{d_i}$$

$$(N-2)! \prod A_i \sum_{d_1+d_2+\ldots+=n-2} \frac{\prod A_i^{d_i}}{\prod d_i!} \prod (d_i+1)$$

注意到 $\sum_{d_1+d_2+\ldots+=n-2} \frac{\prod A_i^{d_i}}{\prod d_i!}$ 其实就是 $\frac{(\sum A_i)^{n-2}}{(n-2)!}$
考虑后面括号内的某一项。组合意义为从 n 个里面选出某 k 个 d_i，那么它对答案的贡献为

$$\prod A_k \frac{(\sum A_i)^{n-2-k}}{(n-2-k)!}$$

dp 算下背包就可以了。

Matvey's Birthday

给一个字符串，字符集大小为 8。
图 G 中,i 与 j 有边当且仅当 $S_i = S_j$ 或者 $|i - j| = 1$。
求图的直径。
$|S| \leq 10^5$。

Matvey's Birthday

显然直径至多 $2|\Sigma| - 1$。
令 $f_{i,c}$ 表示点 i 出发走到字符为 c 的位置的最短距离。可以 BFS。
令 $g_{c1,c2}$ 表示从字符 c_1 走到字符 c_2 最短路。
容易发现 $g_{s_i,c} \leq f_{i,c} \leq g_{s_i,c} + 1$。
点 i 与点 j 的距离 $dis(i,j) = \min(|i-j|, \min(f_{i,c} + 1 + f_{j,c}))$

Matvey's Birthday

从小到大枚举 i，计算 $\max_{j<i} dis(i,j)$
由 $|i-j|$ 贡献的只会是最后几个，之前的贡献都是 $\min(f_{i,c}+1+f_{j,c})$。
而 $g_{s_i,c} \le f_{i,c} \le g_{s_i,c}+1$。每个点到个个字符的距离表示成一个二进制数。
对那些字符相同二进制数相同的位置统计出来的直径是相同的，枚举直接算就可以了。