杂题选讲

starusc

2020 年 10 月 24 日

BZOJ#3716 [PA2014]Muzeum

题目描述

漫展上有 n 件手办和 m 名警卫。每个手办和警卫都可以看做平面直角坐标系上的一个点。警卫们的目光都朝着 y 轴负方向，且都有相同大小的视角（相同的夹角）。警卫可以看见自己视角内（包括边界上的点）的所有手办，不用考虑视线的遮挡。
为了实施抢劫计划，你可以事先贿赂某些警卫。只要某件手办不在任何没被贿赂的警卫的视野内，你就可以偷走它。你知道每件手办的价格，以及每位警卫需要接受多少钱的贿赂。你想知道自己的最大收益是多少。
$n, m \leq 2 * 10^5$

BZOJ#3716 [PA2014]Muzeum

每件手办向可以看见它的警卫连边，那么可以跑一个最大权闭合子图。
复杂度过高，明显过不去。

BZOJ#3716 [PA2014]Muzeum

每件手办向可以看见它的警卫连边，那么可以跑一个最大权闭合子图。复杂度过高，明显过不去。

考虑优化，先将每个警卫的可视范围调整为 90°。假设每个警卫的视角的一半的正切值是 $\frac{w}{h}$。那么一个警卫 (x_1, y_1) 能看见一个手办 (x_2, y_2) 的条件为 $\frac{|x_1-x_2|}{y_1-y_2} \leq \frac{w}{h}, y_1 > y_2$，移项后把所有横坐标乘 h 纵坐标乘 w 即可。然后 $(x, y) \to (x+y, x-y)$ 将视角转为左上角。

BZOJ#3716 [PA2014]Muzeum

每件手办向可以看见它的警卫连边，那么可以跑一个最大权闭合子图。复杂度过高，明显过不去。

考虑优化，先将每个警卫的可视范围调整为 90°。假设每个警卫的视角的一半的正切值是 $\frac{w}{h}$。那么一个警卫 (x_1, y_1) 能看见一个手办 (x_2, y_2) 的条件为 $\frac{|x_1 - x_2|}{y_1 - y_2} \leq \frac{w}{h}, y_1 > y_2$，移项后把所有横坐标乘 h 纵坐标乘 w 即可。然后 $(x, y) \to (x + y, x - y)$ 将视角转为左上角。

最小割 $=$ 最大流，我们考虑模拟求最大流的过程。把坐标按横坐标从小到大排序，依次加入警卫后，把横坐标小于当前警卫的手办加入。一个很显然的贪心，流量肯定是来自纵坐标大于当前警卫纵坐标的最小纵坐标的手办。用 set 维护一下当前还有流量的手办即可。

BZOJ#3716 [PA2014]Muzeum

每件手办向可以看见它的警卫连边，那么可以跑一个最大权闭合子图。复杂度过高，明显过不去。

考虑优化，先将每个警卫的可视范围调整为 90°。假设每个警卫的视角的一半的正切值是 $\frac{w}{h}$。那么一个警卫 (x_1, y_1) 能看见一个手办 (x_2, y_2) 的条件为 $\frac{|x_1-x_2|}{y_1-y_2} \le \frac{w}{h}, y_1 > y_2$，移项后把所有横坐标乘 h 纵坐标乘 w 即可。然后 $(x, y) \to (x+y, x-y)$ 将视角转为左上角。

最小割 $=$ 最大流，我们考虑模拟求最大流的过程。把坐标按横坐标从小到大排序，依次加入警卫后，把横坐标小于当前警卫的手办加入。一个很显然的贪心，流量肯定是来自纵坐标大于当前警卫纵坐标的最小纵坐标的手办。用 set 维护一下当前还有流量的手办即可。

时间复杂度 $O(n \log n)$。

LOJ#6171 记忆的轮廓

题目描述

给你一棵以 1 为根节点的树，有 m 个节点，1 到 n 的简单路径是一条长度为 n 的链，编号递增。

你需要从 1 号点走到 n 号点结束，在 $[1, n]$ 内可以设置 p 个存档位置，1 和 n 必须是存档位置。每次位于树上一个节点时，都会等概率选择一个儿子走下去。每当走到叶子时，再走一步就会读档。

具体的，每次到达一个新的存档位置，存档点便会更新为这个位置（假如现在的存档点是 i，现在走到了一个存档位置 $j > i$，那么存档点便会更新为 j）。读档的意思就是回到当前存档点。

问最优情况下结束路程的期望步数是多少?

$50 \le p \le n \le 500, m \le 1500$ 数据组数 $T \le 5$

LOJ#6171 记忆的轮廓

设 $f_{i,j}$ 表示从 i 出发，当前已经设置了 j 个存档位置了，到 n 的最优期望步数。

LOJ#6171 记忆的轮廓

设 $f_{i,j}$ 表示从 i 出发，当前已经设置了 j 个存档位置了，到 n 的最优期望步数。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

$g_{i,k}$ 表示从 i 到 k 中途没有存档点期望走的步数。

LOJ#6171 记忆的轮廓

设 $f_{i,j}$ 表示从 i 出发，当前已经设置了 j 个存档位置了，到 n 的最优期望步数。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

$g_{i,k}$ 表示从 i 到 k 中途没有存档点期望走的步数。

$$g_{i,j} = g_{i,j-1} + 1 + \frac{0 + \sum_{v>n\&\&v\in son_{j-1}} h_v + gi, j}{|son_{j-1}|}$$

h_v 表示从 v 开始读档的期望步数。

LOJ#6171 记忆的轮廓

设 $f_{i,j}$ 表示从 i 出发，当前已经设置了 j 个存档位置了，到 n 的最优期望步数。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

$g_{i,k}$ 表示从 i 到 k 中途没有存档点期望走的步数。

$$g_{i,j} = g_{i,j-1} + 1 + \frac{0 + \sum_{v>n\&\&v\in son_{j-1}} h_v + gi,j}{|son_{j-1}|}$$

h_v 表示从 v 开始读档的期望步数。

$$h_i = 1 + \frac{\sum_{v\in son_i} h_v}{|son_i|}$$

LOJ#6171 记忆的轮廓

$$g_{i,j} = |son_{j-1}| * (g_{i,j-1} + 1) + \sum_{v>n\&\&v\in son_{j-1}} h_v$$

LOJ#6171 记忆的轮廓

$$g_{i,j} = |son_{j-1}| * (g_{i,j-1} + 1) + \sum_{v>n\&\&v\in son_{j-1}} h_v$$

但是我们求解 f 是 $O(n^3)$，考虑优化 DP。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

LOJ#6171 记忆的轮廓

$$g_{i,j} = |son_{j-1}| * (g_{i,j-1} + 1) + \sum_{v>n\&\&v\in son_{j-1}} h_v$$

但是我们求解 f 是 $O(n^3)$，考虑优化 DP。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

第一种方法，直接 WQS 二分即可。时间复杂度 $O(n^2 \log A)$

LOJ#6171 记忆的轮廓

$$g_{i,j} = |son_{j-1}| * (g_{i,j-1} + 1) + \sum_{v>n\&\&v\in son_{j-1}} h_v$$

但是我们求解 f 是 $O(n^3)$，考虑优化 DP。

$$f_{i,j} = min_{k\in[i+1,n]}\{f_{k,j-1} + g_{i,k}\}$$

第一种方法，直接 WQS 二分即可。时间复杂度 $O(n^2 \log A)$

第二种方法，容易发现决策单调性（若存在位置 $i < j$ 且他们的决策点 $p_i > p_j$，那么让 $p_i = p_j$ 不会使答案更劣)。那么直接用队列维护一下决策三元组即可。时间复杂度 $O(n^2 \log n)$。

LOJ#6669 Nauuo and Binary Tree

题目描述

有一棵 n 个节点的二叉树，根节点为 1，你可以通过询问两个点之间简单路径的距离来还原这棵树，你需要用不超过 30000 次询问求出 $2 \to n$ 号点的父亲。

$n \leq 3000$

LOJ#6669 Nauuo and Binary Tree

首先询问出所有点的深度，按深度从小到大扩展。

每次为一个点找父亲时，我们期望用 $O(\log n)$ 次询问，考虑树链剖分。

LOJ#6669 Nauuo and Binary Tree

首先询问出所有点的深度，按深度从小到大扩展。

每次为一个点找父亲时，我们期望用 $O(\log n)$ 次询问，考虑树链剖分。

每次处理一个点前，把整棵树树链剖分。从根节点所在的重链开始，每次询问当前节点与重链底的距离后，就可以求出它们 LCA 的深度，如果这个深度等于当前点的深度减一，那么我们就找到父亲了，否则跳到 LCA 的轻儿子继续询问。

LOJ#6669 Nauuo and Binary Tree

首先询问出所有点的深度，按深度从小到大扩展。

每次为一个点找父亲时，我们期望用 $O(\log n)$ 次询问，考虑树链剖分。

每次处理一个点前，把整棵树树链剖分。从根节点所在的重链开始，每次询问当前节点与重链底的距离后，就可以求出它们 LCA 的深度，如果这个深度等于当前点的深度减一，那么我们就找到父亲了，否则跳到 LCA 的轻儿子继续询问。

时间复杂度 $O(n^2)$，询问次数 $O(n \log n)$。

CF780H Intranet of Buses

题目描述

平面上有 n 个点，编号相邻的点之间有连边，1 号点和 n 号点之间有连边，形成一个环路，总长为 S。
有 m 个动点从 1 号点出发，速度均为 1 单位长度，第 i 号点在 $\frac{iS}{m}$ 时刻出发。
在点全部出发后，求出编号相邻的点，1 和 n 之间的点的最大距离的最小值。
$n, m \leq 10^5$

CF780H Intranet of Buses

容易发现 x 坐标与 y 坐标都是关于时间 t 的分段一次函数 $x(t), y(t)$，段数有 $O(n)$ 段。

CF780H Intranet of Buses

容易发现 x 坐标与 y 坐标都是关于时间 t 的分段一次函数 $x(t), y(t)$，段数有 $O(n)$ 段。

那么 $f(t) = sqr(x(t) - x(t+c)) + sqr(y(t) - y(t+c)), c = \frac{iS}{m}$，即相邻两个动点间的距离关于时间 t 的分段二次函数，容易发现段数也是 $O(n)$ 段的。

CF780H Intranet of Buses

容易发现 x 坐标与 y 坐标都是关于时间 t 的分段一次函数 $x(t), y(t)$，段数有 $O(n)$ 段。

那么 $f(t) = sqr(x(t) - x(t + c)) + sqr(y(t) - y(t + c)), c = \frac{iS}{m}$，即相邻两个动点间的距离关于时间 t 的分段二次函数，容易发现段数也是 $O(n)$ 段的。

考虑二分答案 ans，对于 $f(t)$ 的每一段求出距离小于 ans 的区间。然后把所有的区间都映射到 $[0, c)$ 内，若其中某个点被覆盖了 m 次，就是合法的了。

CF780H Intranet of Buses

容易发现 x 坐标与 y 坐标都是关于时间 t 的分段一次函数 $x(t), y(t)$，段数有 $O(n)$ 段。

那么 $f(t) = sqr(x(t) - x(t+c)) + sqr(y(t) - y(t+c)), c = \frac{iS}{m}$，即相邻两个动点间的距离关于时间 t 的分段二次函数，容易发现段数也是 $O(n)$ 段的。

考虑二分答案 ans，对于 $f(t)$ 的每一段求出距离小于 ans 的区间。然后把所有的区间都映射到 $[0, c)$ 内，若其中某个点被覆盖了 m 次，就是合法的了。

时间复杂度 $O(n \log A)$。

CF1037G A Game on Strings

题目描述

你有一个只包含小写字母的字符串 s，有两个人不断进行以下操作，无法操作的人输。

- 选一个字符，把字符串中所有这个字符删去，剩下的每一段会变成一个新的独立的字符串。

Q 次询问 $[l, r]$ 表示初始字符串是子串 $s[l, r]$，问是否有必胜策略。

$|s|, Q \leq 10^5$

CF1037G A Game on Strings

考虑枚举每次第一个删除的字符，那么剩下的 SG 函数值就是每个新字符串的 SG 函数的异或值。对于每个字符，考虑预处理出删除这个字符后，新字符串的 SG 前缀异或和。那么询问时中间的区间就可以直接求出，我们只用求两端零散的区间的值即可。

CF1037G A Game on Strings

考虑枚举每次第一个删除的字符，那么剩下的 SG 函数值就是每个新字符串的 SG 函数的异或值。对于每个字符，考虑预处理出删除这个字符后，新字符串的 SG 前缀异或和。那么询问时中间的区间就可以直接求出，我们只用求两端零散的区间的值即可。

考虑如何快速求出每个区间的 SG 值，我们考虑记忆化搜索，记 $f_{0/1,i,j}$ 表示从 i 开始向左或向右到第一个字符 j 前，这个区间的 SG 值。这样我们就只有 $26n$ 个有效区间。

CF1037G A Game on Strings

考虑枚举每次第一个删除的字符，那么剩下的 SG 函数值就是每个新字符串的 SG 函数的异或值。对于每个字符，考虑预处理出删除这个字符后，新字符串的 SG 前缀异或和。那么询问时中间的区间就可以直接求出，我们只用求两端零散的区间的值即可。

考虑如何快速求出每个区间的 SG 值，我们考虑记忆化搜索，记 $f_{0/1,i,j}$ 表示从 i 开始向左或向右到第一个字符 j 前，这个区间的 SG 值。这样我们就只有 $26n$ 个有效区间。

时间复杂度 $O(26^2n + 26Q)$。

CFgym102412#B Alexey the Sage of The Six Paths

题目描述

一个左边 n 个点，右边 n 个点的二分图（初始没有边）。你需要在其中插入 m 条边。若第 i 个点的度数是 c_i，那么需要花费 p_{i,c_i} 的代价。给定 l, r，求出完全匹配在 $[l, r]$ 内的最小代价。

$n, m \leq 30$

CFgym102412#B Alexey the Sage of The Six Paths

如果我们确定了每一个点的度数，还有最大匹配的最大值与最小值，那么在这个区间内的最大匹配都可以取到。(每次在左右各找一个度数不为 0 的非匹配点 u, v ，假设它们连向 p, q （p, q 显然为匹配点)，然后删掉 $(u,p),(v,q)$ 加上 $(p,q),(u,v)$ 最大匹配数就加一了)

CFgym102412#B Alexey the Sage of The Six Paths

如果我们确定了每一个点的度数，还有最大匹配的最大值与最小值，那么在这个区间内的最大匹配都可以取到。(每次在左右各找一个度数不为 0 的非匹配点 u, v ，假设它们连向 p, q （p, q 显然为匹配点)，然后删掉 $(u, p), (v, q)$ 加上 $(p, q), (u, v)$ 最大匹配数就加一了)

根据 Hall 定理，最大匹配数等于 $|S| - max_{S' \subseteq S}(|S'| - |tr(S')|)$，其中 $tr(S')$ 表示 S' 的相邻的点集。

CFgym102412#B Alexey the Sage of The Six Paths

如果我们确定了每一个点的度数，还有最大匹配的最大值与最小值，那么在这个区间内的最大匹配都可以取到。(每次在左右各找一个度数不为 0 的非匹配点 u, v ，假设它们连向 p, q （p, q 显然为匹配点)，然后删掉 $(u, p), (v, q)$ 加上 $(p, q), (u, v)$ 最大匹配数就加一了)

根据 Hall 定理，最大匹配数等于 $|S| - max_{S' \subseteq S}(|S'| - |tr(S')|)$，其中 $tr(S')$ 表示 S' 的相邻的点集。

要使最大匹配的最小值小于等于 r，只需要存在一个子集 $|S| + |tr(S')| - |S'| \leq r$。要使最大匹配的最大值大于等于 l，只需要两边都至少有 l 个度数非零的点。

CFgym102412#B Alexey the Sage of The Six Paths

考虑 DP，$f_{i,a,b,c,d}$，表示 $[1,i]$ 的点中，a 个在我们选定的点集，选定的点集的度数和为 b，度数大于 0 的点的个数为 c，总共用了 d 的度数的最小代价。枚举 $i+1$ 号点的度数，是否在点集内进行转移。

CFgym102412#B Alexey the Sage of The Six Paths

考虑 DP，$f_{i,a,b,c,d}$，表示 $[1,i]$ 的点中，a 个在我们选定的点集，选定的点集的度数和为 b，度数大于 0 的点的个数为 c，总共用了 d 的度数的最小代价。枚举 $i+1$ 号点的度数，是否在点集内进行转移。

两半分别 DP 后，枚举两边的点集大小，点集度数，非零点个数，来算答案。最后按我们上述所讲构造方案即可。

CFgym102412#B Alexey the Sage of The Six Paths

考虑 DP，$f_{i,a,b,c,d}$，表示 $[1,i]$ 的点中，a 个在我们选定的点集，选定的点集的度数和为 b，度数大于 0 的点的个数为 c，总共用了 d 的度数的最小代价。枚举 $i+1$ 号点的度数，是否在点集内进行转移。

两半分别 DP 后，枚举两边的点集大小，点集度数，非零点个数，来算答案。最后按我们上述所讲构造方案即可。

时间复杂度 $O(n^3m^3)$。

CF936E Iqea

题目描述

给定二维网格图上有不同的 n 个格子，相邻的格子连通且距离为 1。
保证所有格子连通，且网格图的补图连通。
一共 m 个操作，分为两种：

- 在一个点上建一个商店
- 询问离一个点最近的商店的距离

$n, m, x_i, y_i \leq 3 * 10^5$

CF936E Iqea

补图连通就是不存在一个环中还有空白格点。据此我们可以对这个网格图做一个转换。把每列的列联通块（连续的块）看做一个点，这样图就变成了一棵树。

CF936E Iqea

补图连通就是不存在一个环中还有空白格点。据此我们可以对这个网格图做一个转换。把每列的列联通块（连续的块）看做一个点，这样图就变成了一棵树。

求最近距离是一个经典的点分树问题。考虑两个点 a, b 它们的最短路径经过了 C 这个块那么最短距离为 $d_a + d_b + |c_a - c_b|$，d 表示到 C 的最短距离，c 表示点经过最短路径到 C 时的纵坐标。

CF936E Iqea

补图连通就是不存在一个环中还有空白格点。据此我们可以对这个网格图做一个转换。把每列的列联通块（连续的块）看做一个点，这样图就变成了一棵树。

求最近距离是一个经典的点分树问题。考虑两个点 a, b 它们的最短路径经过了 C 这个块那么最短距离为 $d_a + d_b + |c_a - c_b|$，d 表示到 C 的最短距离，c 表示点经过最短路径到 C 时的纵坐标。

把绝对值拆开后，发现我们求的其实是一个前缀 min 和后缀 min，在每个块上开一个树状数组维护即可。

CF936E Iqea

补图连通就是不存在一个环中还有空白格点。据此我们可以对这个网格图做一个转换。把每列的列联通块（连续的块）看做一个点，这样图就变成了一棵树。

求最近距离是一个经典的点分树问题。考虑两个点 a, b 它们的最短路径经过了 C 这个块那么最短距离为 $d_a + d_b + |c_a - c_b|$，d 表示到 C 的最短距离，c 表示点经过最短路径到 C 时的纵坐标。

把绝对值拆开后，发现我们求的其实是一个前缀 min 和后缀 min，在每个块上开一个树状数组维护即可。

时间复杂度 $O(n \log^2 n)$

CF981H K Paths

题目描述

给你一颗 n 个节点的树，你可以自选 k 条简单路径（选的路径可以重复，且具有先后顺序），满足树上每条边要么被 0 或 1 或 k 条路径覆盖，且必须存在一条边被 k 条路径覆盖，求选择路径的方案数，答案对 998244353 取模。

$n, k \le 10^5$

CF981H K Paths

被覆盖 k 次的边一定组成了一条链 $u \to v$，容易发现我们可以算在 u, v 的子树中各选 k 个的点的方案数，然后相乘即可。

CF981H K Paths

被覆盖 k 次的边一定组成了一条链 $u \to v$，容易发现我们可以算在 u, v 的子树中各选 k 个的点的方案数，然后相乘即可。

u 的每个儿子子树中只能选一个儿子，易得生成函数为

$$\prod_{p \in son_u} (1 + siz_p x)$$

CF981H K Paths

被覆盖 k 次的边一定组成了一条链 $u \to v$，容易发现我们可以算在 u, v 的子树中各选 k 个的点的方案数，然后相乘即可。

u 的每个儿子子树中只能选一个儿子，易得生成函数为

$$\prod_{p \in son_u} (1 + siz_p x)$$

注意到路径可重，所以我们可以选小于 k 条路径，剩下的用 u 来填充。一个点的答案为（$k = 1$ 时需要特判），$f_u = \sum_{i=0}^{k} \frac{k!}{(k-i)!} a_i$， a_i 为上述生成函数第 i 项的系数

CF981H K Paths

但如果 u, v 是祖先儿子关系时就不能这么算。考虑 u 是 v 祖先的情况。p 是 u 儿子，v 的祖先。

$$f_v \sum_{i=0}^{k} \frac{k!}{(k-i)!}[x^i](\frac{1+(n-siz_u)x}{1+siz_p x}) \prod_{w \in son_u} (1+siz_w x)$$

CF981H K Paths

但如果 u, v 是祖先儿子关系时就不能这么算。考虑 u 是 v 祖先的情况。p 是 u 儿子，v 的祖先。

$$f_v \sum_{i=0}^{k} \frac{k!}{(k-i)!}[x^i](\frac{1+(n-siz_u)x}{1+siz_p x}) \prod_{w \in son_u}(1+siz_w x)$$

把 p 的子孙放一起后就可以分治 NTT 了

$$\sum_{i=0}^{k} \frac{k!}{(k-i)!}[x^i](1+(n-siz_u)x) \prod_{p \in son_u}((\sum_{v} f_v) \prod_{w \neq p}(1+siz_w x))$$

CF981H K Paths

但如果 u,v 是祖先儿子关系时就不能这么算。考虑 u 是 v 祖先的情况。p 是 u 儿子，v 的祖先。

$$f_v\sum_{i=0}^{k}\frac{k!}{(k-i)!}[x^i](\frac{1+(n-siz_u)x}{1+siz_px})\prod_{w\in son_u}(1+siz_wx)$$

把 p 的子孙放一起后就可以分治 NTT 了

$$\sum_{i=0}^{k}\frac{k!}{(k-i)!}[x^i](1+(n-siz_u)x)\prod_{p\in son_u}((\sum_{v}f_v)\prod_{w\neq p}(1+siz_wx))$$

时间复杂度 $O(n\log^2 n)$。

谢谢大家。