THUPC2019 解题报告

A

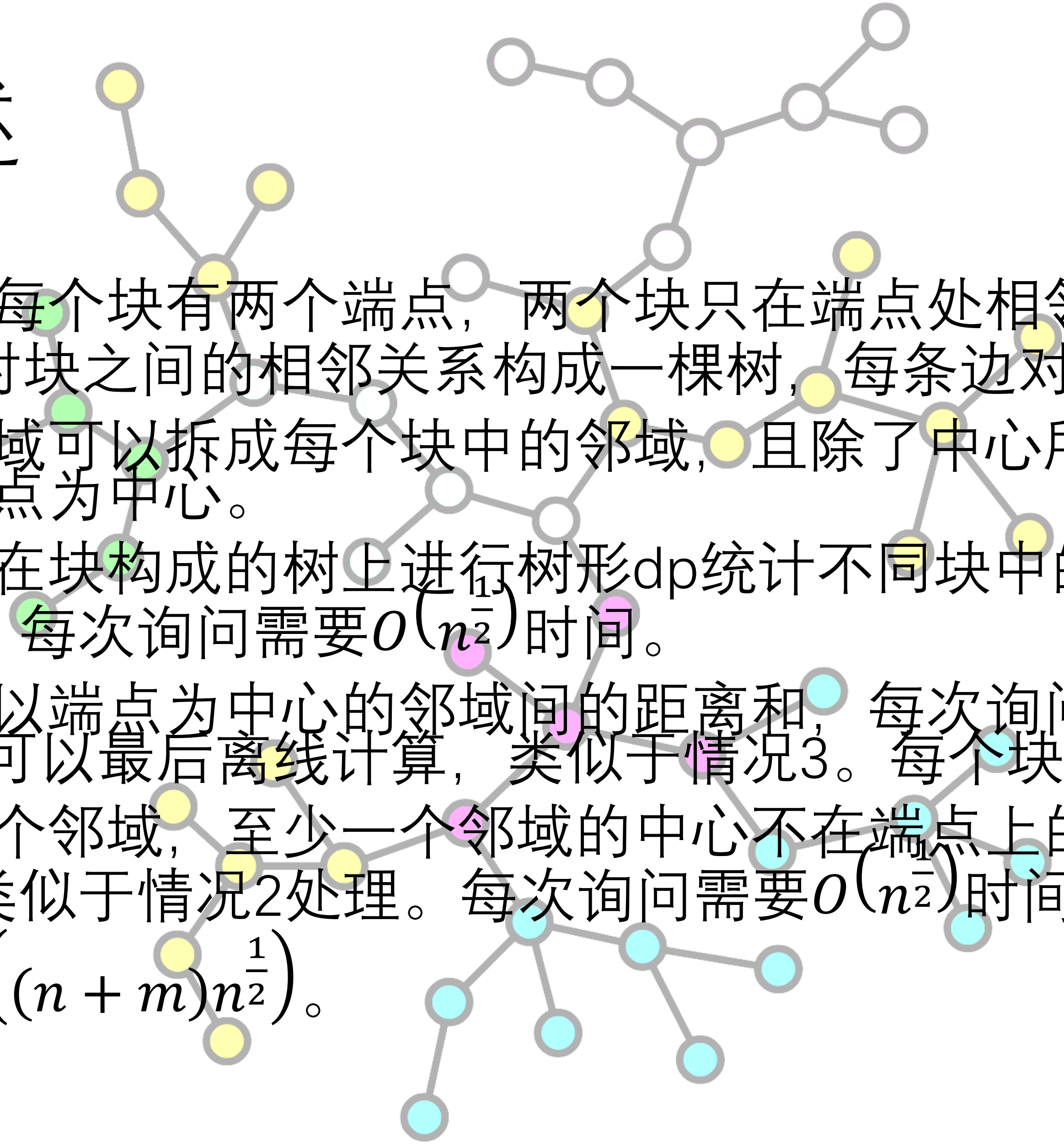
简要题意

- 树上的一个邻域定义为到点 x 距离不超过 y 条边的点集，x 称为邻域的中心，y 称为邻域的半径。
- 给一棵 n 个点的树，m 次询问，每次给出两个邻域，问两个邻域中各取一个点，两两点对间的距离之和 。
- $n, m \leq 100000$

特殊限制下的算法

- 情况1：如果两个邻域不相交，则存在一个点c，使得两个邻域间任意路径经过c，只需统计两个邻域的点数以及到c的距离和。
- 情况2：只有一次询问。
- d(a,b)=d(a,root)+d(b,root)-2d(lca(a,b),root)
- 转化为求两两点间lca的深度和。将第一个邻域中 的每个点到根的路径上边权加上1，求第二个邻域中的每个点到根的路径的边权和。时间复杂度$O(n)$。
- 情况3：所有询问中，两个邻域的中心是固定的两个点x,y，但半径可能不同。
- 枚举第一个邻域的半径，类似情况2处理，处理出所有可能的询问的答案，时间复杂度$O(n^2)$。

解法简述



- 树分块，要求每个块有两个端点，两个块只在端点处相邻，块数和块大小均为$O\left(n^{\frac{1}{2}}\right)$。此时块之间的相邻关系构成一棵树，每条边对应一个块。
- 原树的一个邻域可以拆成每个块中的邻域，且除了中心所在的块，其余块的邻域以块的端点为中心。
- 每次询问时，在块构成的树上进行树形dp统计不同块中的邻域之间的距离和，类似于情况1。每次询问需要$O\left(n^{\frac{1}{2}}\right)$时间。
- 同个块内两个以端点为中心的邻域间的距离和，每次询问在每个块中产生O(1)个询问，可以最后离线计算，类似于情况3。每个块需要O(n)时间。
- 同个块内的两个邻域，至少一个邻域的中心不在端点上的情况，每次询问至多出现2次，类似于情况2处理。每次询问需要$O\left(n^{\frac{1}{2}}\right)$时间。
- 时间复杂度$O\left((n+m)n^{\frac{1}{2}}\right)$。

B

简要题意

- 给定一组数据结构的操作，判断该数据结构是否可能是栈、队列、大根堆、小根堆中的某一种

- 数据规模：$n \le 10^5$

解法简述

- 直接模拟即可

- 坑1：输出的值比加入的值要多
- 坑2：-2147483648

- 时间复杂度：$O(n\log n)$

C

题目简述

- 给定一个二元组集合 S，定义二维数列 f：
- $f_{n,m} = \begin{cases} 1 & (n,m) = (x,1)或(1,y) \\ f_{n-1,m} + f_{n,m-1} + f_{n-1,m-1} & n,m \neq 1 \\ 0 & (n,m) \in S \end{cases}$
- 给定 n,m，求$2\left(\sum_{j=1}^{m} f_{n,j} + \sum_{i=1}^{n} f_{i,m}\right) - f_{n,m}$

没有障碍的情况

- 求得f 的 OGF 为$f(x,y)=\frac{1}{1-x-y-xy}$，且，只需要乘上 $\frac{1}{1-x}$ 或 $\frac{1}{1-y}$ 就能得到某一行或某一列的前缀和的 OGF。展开之后，显然可以在 $O(m)$ 的时间内求出某个 $f_{n,m}$ 或 f 某一列或某一行的前缀和。
- 注意到模数是质数，使用 Lucas 定理求二项式系数的值。

有障碍的情况

- 考虑容斥。设 g_i 为第一次走到的障碍为第 i 个障碍的方案数。先求出经过 i 的总方案数，再枚举位于 i 的最下方的障碍 j 减去那些方案 g_j，就可得到 g_i。实现上可以通过建出 DAG 后拓扑排序递推求解。
- 总时间复杂度为 $O(k^2m)$

D

题目大意

- 在圆周上选n个点，两两之间连线。
- 问这些弦最多能把圆内分成多少份。
- $n \leq 64$

解法

- 考虑在x个点（x>2）的图中新加入一个点
- 这个点会导致答案增加的数量为:
- $x-1+C_{x-1}^{3}$
- 递推即可
- 答案在int以内

E

简要题意

- 空间中有两个物体，物体 A 是一个凸多面体，物体 B 是最多 4 个凸多面体的并。
- 物体 A 每秒钟均匀移动一个向量 v。
- 令 $f(t)$ 为 t 时刻两个物体的交的体积，求$\int_0^\infty f(t)$.

解法简述

- 首先可以注意到通过容斥原理，可以将问题转化为给 B 只由一个凸多面体组成的情况。
- 注意到 A 和 B 的交的凸多面体上的点只会在一些关键点上变化，并且在关键点之间，交的体积是一个时间的三次函数。
- 求出所有关键点，然后每个关键点之间进行积分就可以了。

解法简述 Cont.

- 为此需要实现凸多面体的交。
- 可以用一个面的 list 来表示一个凸多面体，然后实现实现一个 convexCut 函数来表示用一个给定的面去切一个给定的凸多面体的话，切出来的凸多面体。
- 实现了这个函数之后就可以求两个凸多面体的交了。
- 从输入给定的点中可以用暴力的 n^4 枚举来找出凸包上的所有的面。
- 也需要实现其它一些三维几何基本操作。

F

题意

- k 维空间，每维坐标是 $[0, n)$ 的整数
- 设 $N = n^k$，输入所有 n^k 个点的权值
- q 次独立、**在线**的询问，每次给出一个起始点 A 与自然数 T:
  - 重复 T 次，每次可以将当前位置移动到恰有一维不同的另一个点上
  - 询问所有可能的最终位置（不去重）的权值之和，对 P 取模

- 数据规模：$N \le 10^6$，$q \le 5 \times 10^5$，$P = 998244353$，每次询问 $T < P$

性质

- 性质0： $k = O(\log N)$
- 性质1： 在询问中，只要 T 一定，每个最终位置被计算的次数只取决于该位置到初始位置的距离，而与具体的坐标无关
  - 两个点的距离定义为坐标不同的维度数

- 可将每个询问拆成以下两个问题：
- 问题1：给定起始点 A，对所有的 $d \in [0, k]$，询问与点 A 距离为 d 的点权值之和
- 问题2：给定 T，对所有的 $d \in [0, k]$，询问在 T 次操作后每个距离为 d 的点被计算的次数

解法

- 问题1：询问与点 A 距离为 d 的点权值之和
  - 动态规划，设 $f(A,i,j)$ 表示：与点 A 坐标在前 i 维中恰好有 j 维不同，在其他维度均相同的所有点的权值之和
  - 时间 $O(N\log^2 N + q\log N)$，滚动数组后空间 $O(N\log N)$
- 问题2：询问在 T 次操作后每个距离为 d 的点被计算的次数
  - 可以改为计算所有距离为 d 的点被计算的总次数，这是线性递推问题，除以一个组合数即可得到答案
  - 朴素线性递推时间 $O(q\log^3 N\log P)$，可以通过预处理分段打表做到时间 $O\left(\sqrt{P}\log^{2.5} N + q\log^2 N\right)$，空间 $O\left(\sqrt{P}\log^{1.5} N\right)$
- （时间限制是 4 倍标程用时）

G

简要题意

- 有一个序列 S，A 和 B 玩游戏：
- 每次 A 给出一个序列 T, B 把 T 循环移位之后加到 S 上。
- 如果某一步结束 S 每个数都是给定质数 P 的倍数，那么 A 胜利
- 问 A 能否胜利，如果能，最少几步胜利。

解法

- 这里不加证明地给出：
- Prop 1. 当且仅当 $N = P^k$，先手对任意的 S 必胜。如果 $N = P^k \cdot r$，那么只有 S[i] = S[i + p^k] 时先手必胜。
  - 以下只考虑 $N = P^k$ 的情况。

- 我们在 Z_p 上构造一个 $P^k \times P^k$ 的矩阵 A：
  - $A(0,0) = 1, A(0,i) = 0, \forall i > 0$
  - $A(i,j) = A(i-1,j-1) + A(i-1,j)$
- 事实上，$A(i,j) = (-1)^{i-j} C(i,j)$

解法 cont’d

- Prop 2. 把 S 写成行向量 s，求解方程：$x A = s$，得到的 x 里取出最小的 i 使得 $x_i \neq 0$，那么 $P^k - i$ 即为答案。
  - 证明略
- 使用二项式反演，上述方程等价于 $x = s \cdot C$，其中 C 为组合数矩阵。最后，由 Lucas 定理知道 $C(i,j)$ 只有 i 在 p 进制下每位大于等于 j 才非零，就能使用 Mobius 变换类似的方法计算。
- 总时间复杂度 $O(NP\log N)$

H

简要题意

- 根据鸭棋的规则，要求模拟整个棋局

解法简述

- 大模拟
- 一个 trick 是，如果我们规定广义骑士的走子规则为合法方向向量如下的棋子：$(\pm a, \pm b), (\pm a, \mp b), (\pm b, \pm a), (\pm b, \mp a)$。则其移动时棋盘上不能有障碍的位置（以$(a, b)$为例）为$\left(\frac{ka}{\max\{a,b\}}, \frac{kb}{\max\{a,b\}}\right)$，其中$k = 1, \dots, \max\{a, b\}$。
- 不难发现，所有棋子本质上都是广义骑士。
- 可以省去写走子规则的很多代码。

简要题意

- 给 n 个形如 $f_i(x) = |a_i x + b_i|$ 的函数，求这些函数的每个前缀和的最大值
- $n \le 5 \times 10^5$

简单情况

- 先考虑对于单个 $f(x) = \sum_{i=1}^{n} |a_i x + b_i|$ 如何求其最小值
- 根据高考不等式的知识可知，$f(x)$ 是一个形如下凸包的函数，每个凸点都对应 $f(x)$ 每一项的零点。将这些零点排序，找到凸包斜率由负变正的点（即凸包的下顶点）即可。

最终解法

- 于是便有了一个按照输入顺序依次插入点，按照 $-\frac{a_i}{b_i}$ 的值作为横坐标维护下凸包，每次寻找凸包的下顶点的做法。只需要维护 a_i 的前缀和，二分找到斜率变号的位置即可。
- 使用树状数组或者线段树都可以完成。

J

简要题意

- 给定一张有向图，每个节点有颜色和点权。
- 保证所有不同的颜色为 1 的节点之间不互达。
- 要求选出一些（至少一个）颜色为 1 的节点，使得所有这些节点、及它们直接或间接到达的节点中：**权值和最大的其他颜色节点的权值和**与 **1 号颜色节点的权值和**的比值最大。

- 数据规模：$n \le 700$，$m \le 6000$

解法简述

- 不难发现，一定存在最优策略只选了一个颜色为 1 的节点（可用反证法证）。
- 于是传递闭包后枚举选择的节点并暴力计算答案即可。

- 时间复杂度：$O(n^2)$

K

简要题意

- 定义一种按位独立的二进制运算
- 每一位分别是与、或和异或中的一种
- 给出一张图，求在这种运算的和的意义下的最大生成树
- $n \le 70, m \le 5000$，二进制位数 $W \le 12$

解法简述

- 这道题数据范围很小，可把这道题转化为计数题，即对于 $i \in [0, 2^W - 1]$，统计权值为 i 的树的方案数
- 生成树计数显然基尔霍夫矩阵，但考虑权值就很麻烦了（除法）
- 注意到，如果所有操作都是异或，那我们可以
  - 将矩阵中的每个权值集合进行 FWT
  - FWT 后数组（集合幂级数）中的每一位都独立了
  - 可以分每一位分别计算基尔霍夫矩阵去掉一行一列后的行列式
  - 把行列式的答案组成一个新的数组（集合幂级数）
  - 将它 IFWT 后即可得到每一个权值对应的树的方案数
- 与和或的运算也可以用 FMT 达到一样的效果

解法简述 cont.d

- 注意到
  - FWT 的本质是每一个二进制位分别 FFT（长度为 2 的 FFT）
  - FMT 的本质是每一个二进制位分别前/后缀和
- 那我们可以将这两种算法结合起来
  - 如果某一二进制位的操作符是异或，那么我们就将这一位进行 FFT
  - 如果某一二进制位的操作符是与（或），就将这一位求后（前）缀和
- 这样就转化为按（数组里的）位独立的问题了，从而可以按照上面的方法分别计算行列式，最后再拼起来，按照这样的规则逆变换回去
- 注意到我们只需要判断 0 和非 0，取几个质数在模意义下计算行列式即可，复杂度 $O\left((n+W)n^2 2^W\right)$

L

简要题意

- 给定 m 个可重集合，第 i 个的大小为 n_i。
- 求它们两两合并后的 m^2 个集合的中位数，结果哈希输出。

- 数据规模：$m \le 10000$，$n_i \le 500$

解法简述

- 考虑集合 $\mathrm{A}=\{a_1, a_2, \dots, a_s\}$ 和 $\mathrm{B}=\{b_1, b_2, \dots, b_t\}$，方便起见我们假设 $a_1, \dots, a_s$ 和 $b_1, \dots, b_t$ 均升序，且 s, t 均为偶数（其他情况也有类似结论和解法，这里不详细讨论）。
- 不妨假设它们合并后中位数为 a_j，且有 k 满足 $b_{k+1} \geq a_j \wedge b_k \leq a_j$，则不难发现 j, k 满足 $k+j=\frac{s+t}{2}$，即 $\left(\frac{s}{2}-j\right)+\left(\frac{t}{2}-k\right)=0$。
- 考虑所有 $\sum n_i$ 个元素作为中位数的所有可能：不难发现只需要将所有元素排序并升序考虑所有 $A_{i,j}$，即可借助链表维护相对当前 $A_{i,j}$ 的 $\frac{t}{2}-k$ 值对应的所有序列，快速查询即可求解。

复杂度分析

- 对于 $A_{i,j}$ 的考虑，由于只会求出中位数为 $A_{i,j}$ 的集合对，因此总体上只有所有 m^2 对集合对的答案会被计算，且它们都只会被恰好计算一次。
- 排序可以使用基数排序优化。

- 综上，时间复杂度为 $O(m^2 + m\sum n_i)$。

M

简要题意

- 给定一个年份，求该年母亲节的日期

解法简述

- 逐年计算即可。