图论数学串讲

合肥一中 whx

January 9, 2018

前言

- 本来大家将会欣赏到如下两节课:

前言

- 本来大家将会欣赏到如下两节课:
- 数学 without 数论计数

前言

- 本来大家将会欣赏到如下两节课:
- 数学 without 数论计数
- 图论 without 网络流

前言

- 本来大家将会欣赏到如下两节课:
- 数学 without 数论计数
- 图论 without 网络流
- 觉得太奇怪了干脆放到一起来讲吧

前言

- 本来大家将会欣赏到如下两节课:
- 数学 without 数论计数
- 图论 without 网络流
- 觉得太奇怪了干脆放到一起来讲吧
- 假设大家都会了网络流和计数，如果不会可以问讲这个的同学...

图划分

- 来一道小题热身

图划分

- 来一道小题热身
- 有一个 n 个点的图，现在请你求出一个划分，并且在每个划分里找出一个 center

图划分

- 来一道小题热身
- 有一个 n 个点的图，现在请你求出一个划分，并且在每个划分里找出一个 center
- 每个联通块的半径定义为距离 center 最远的点的距离，要求每个联通块至少要有 k 个点

图划分

- 来一道小题热身
- 有一个 n 个点的图，现在请你求出一个划分，并且在每个划分里找出一个 center
- 每个联通块的半径定义为距离 center 最远的点的距离，要求每个联通块至少要有 k 个点
- 最小化最大半径，两倍近似

图划分

- 来一道小题热身
- 有一个 n 个点的图，现在请你求出一个划分，并且在每个划分里找出一个 center
- 每个联通块的半径定义为距离 center 最远的点的距离，要求每个联通块至少要有 k 个点
- 最小化最大半径，两倍近似
- 要求多项式算法

图划分

- 首先选 center，如果选出的 center 都不在一个联通快里，那么直接匹配即可。

图划分

- 首先选 center，如果选出的 center 都不在一个联通快里，那么直接匹配即可。
- 如何保证选出的 center 都不在一个联通块里，并且存在一个合法方案（没有点没有被 2r 覆盖)?

图划分

- 首先选 center，如果选出的 center 都不在一个联通快里，那么直接匹配即可。
- 如何保证选出的 center 都不在一个联通块里，并且存在一个合法方案（没有点没有被 2r 覆盖)?
- 每次找一个没有被覆盖的点做新 center 就好了

石子游戏

- n 个石子，权值从 a_1 到 a_n。

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:
- 把 T 中石子丢掉，$S-T$ 中石子权值减 1

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:
- 把 T 中石子丢掉，$S-T$ 中石子权值减 1
- 把 $S-T$ 中石子丢掉，T 中石子权值减 1

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:
- 把 T 中石子丢掉，$S-T$ 中石子权值减 1
- 把 $S-T$ 中石子丢掉，T 中石子权值减 1
- 如果所有石子都被丢掉了 Bob 胜利

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:
- 把 T 中石子丢掉，$S-T$ 中石子权值减 1
- 把 $S-T$ 中石子丢掉，T 中石子权值减 1
- 如果所有石子都被丢掉了 Bob 胜利
- 如果某时刻有石子权值为 0，那么 Alice 胜利

石子游戏

- n 个石子，权值从 a_1 到 a_n。
- 每次 Alice 选一个真子集 T，Bob 有两个选择:
- 把 T 中石子丢掉，$S-T$ 中石子权值减 1
- 把 $S-T$ 中石子丢掉，T 中石子权值减 1
- 如果所有石子都被丢掉了 Bob 胜利
- 如果某时刻有石子权值为 0，那么 Alice 胜利
- $n \leq 10^6, a_i \leq 10^6$

拉格朗日插值

- 变成 x^i 的和

牛顿插值和差分

- 变成 $C(x,i)$ 的和

向量、矩阵、线性相关

- 向量

向量、矩阵、线性相关

- 向量
- 正交和线性相关

向量、矩阵、线性相关

- 向量
- 正交和线性相关
- 矩阵和矩阵的秩

向量、矩阵、线性相关

- 向量
- 正交和线性相关
- 矩阵和矩阵的秩
- 矩阵乘法和它的不同理解

向量、矩阵、线性相关

- 向量
- 正交和线性相关
- 矩阵和矩阵的秩
- 矩阵乘法和它的不同理解
- 矩阵的行空间列空间和零空间

BZOJ3569 Dzy loves Chinese

- 一个 $n \le 10^5$ 个点的无向图，$q \le 10^5$ 个询问。

BZOJ3569 Dzy loves Chinese

- 一个 $n \le 10^5$ 个点的无向图，$q \le 10^5$ 个询问。
- 每次删掉 $k \le 15$ 条边，询问连通性。

BZOJ3569 Dzy loves Chinese

- 取一棵生成树

BZOJ3569 Dzy loves Chinese

- 取一棵生成树
- 给非树边随机权值，xor 到树边上

BZOJ3569 Dzy loves Chinese

- 取一棵生成树
- 给非树边随机权值，xor 到树边上
- 线性相关？ why?

UEG

- 二分图上不能走过重复的边，每人走一步，谁会获胜？

UEG

- 二分图上不能走过重复的边，每人走一步，谁会获胜?
- 经典问题

UEG

- 二分图上不能走过重复的边，每人走一步，谁会获胜?
- 经典问题
- 显然如果存在一个集合 S，满足外界每个点到 S 的边数都是偶数，那么从 S 里出发，无论先手怎么移动，后手总能一步移动回来，从而此时后手必胜

UEG

- 二分图上不能走过重复的边，每人走一步，谁会获胜?
- 经典问题
- 显然如果存在一个集合 S，满足外界每个点到 S 的边数都是偶数，那么从 S 里出发，无论先手怎么移动，后手总能一步移动回来，从而此时后手必胜
- 二分图上我们不妨试着在一侧找出这样一个 $evenkernel$，显然这就等于是在二分图的邻接矩阵中找一个包含起点的 $xor = 0$ 的向量集合

UEG

- 二分图上不能走过重复的边，每人走一步，谁会获胜?
- 经典问题
- 显然如果存在一个集合 S，满足外界每个点到 S 的边数都是偶数，那么从 S 里出发，无论先手怎么移动，后手总能一步移动回来，从而此时后手必胜
- 二分图上我们不妨试着在一侧找出这样一个 $evenkernel$，显然这就等于是在二分图的邻接矩阵中找一个包含起点的 $xor = 0$ 的向量集合
- 为了证明这一点我们要证明如果不存在一个对第 i 行的线性表出，那么先手就一定可以通过移动一步使得存在一个对第 j 列的线性表出

UEG

- 先手在左侧第 i 个，等于是在第 i 行，每移动一步等于选择第 j 列，然后把 i 行 j 列的 1 变成 0，然后后手就在第 j 列

UEG

- 先手在左侧第 i 个，等于是在第 i 行，每移动一步等于选择第 j 列，然后把 i 行 j 列的 1 变成 0，然后后手就在第 j 列
- 也就是如果不存在对 i 行的线性表出，那么一定存在一个 j，使得 $a[i][j]=0$ 之后存在对 j 这一列的线性表出

UEG

- 先手在左侧第 i 个，等于是在第 i 行，每移动一步等于选择第 j 列，然后把 i 行 j 列的 1 变成 0，然后后手就在第 j 列
- 也就是如果不存在对 i 行的线性表出，那么一定存在一个 j，使得 $a[i][j]=0$ 之后存在对 j 这一列的线性表出
- 也就等价于修改之前列上可以 xor 出一个只有第 i 行是 1 的向量，这一定是合法的

UEG

- 先手在左侧第 i 个，等于是在第 i 行，每移动一步等于选择第 j 列，然后把 i 行 j 列的 1 变成 0，然后后手就在第 j 列
- 也就是如果不存在对 i 行的线性表出，那么一定存在一个 j，使得 $a[i][j]=0$ 之后存在对 j 这一列的线性表出
- 也就等价于修改之前列上可以 xor 出一个只有第 i 行是 1 的向量，这一定是合法的
- 因为我们把它加到矩阵的最后一列，因为只有第 i 行是 1，而第 i 行本来就和别的行线性无关，所以矩阵的秩不会改变

UEG

- 先手在左侧第 i 个，等于是在第 i 行，每移动一步等于选择第 j 列，然后把 i 行 j 列的 1 变成 0，然后后手就在第 j 列
- 也就是如果不存在对 i 行的线性表出，那么一定存在一个 j，使得 $a[i][j]=0$ 之后存在对 j 这一列的线性表出
- 也就等价于修改之前列上可以 xor 出一个只有第 i 行是 1 的向量，这一定是合法的
- 因为我们把它加到矩阵的最后一列，因为只有第 i 行是 1，而第 i 行本来就和别的行线性无关，所以矩阵的秩不会改变
- 也就意味着前面的列可以 xor 出来这个向量，从而证明了这个结论

行列式、特征值、特征多项式

- 行列式

行列式、特征值、特征多项式

- 行列式
- 特征值

行列式、特征值、特征多项式

- 行列式
- 特征值
- 特征多项式、Jordan 标准型

例题 1：集训队集训 2017 Day2t3

- 打 Boss，Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。

例题 1：集训队集训 2017 Day2t3

- 打 Boss，Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。
- 这个“恐怖的奴隶主”有一个特殊的技能：每当它被扣减生命值但没有死亡（死亡即生命值 ≤ 0），且 Boss 的随从数量小于上限 k，便会召唤一个新的具有 m 点生命值的“恐怖的奴隶主”。

例题 1：集训队集训 2017 Day2t3

- 打 Boss，Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。
- 这个“恐怖的奴隶主”有一个特殊的技能：每当它被扣减生命值但没有死亡（死亡即生命值 ≤ 0），且 Boss 的随从数量小于上限 k，便会召唤一个新的具有 m 点生命值的“恐怖的奴隶主”。
- 现在小 Y 可以进行 n 次攻击，每次攻击时，会从 Boss 以及 Boss 的所有随从中的等概率随机选择一个，并扣减 1 点生命值，她想知道进行 n 次攻击后扣减 Boss 的生命值点数的期望。为了避免精度误差，你的答案需要对 998244353 取模。

例题 1：集训队集训 2017 Day2t3

- 打 Boss，Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。
- 这个“恐怖的奴隶主”有一个特殊的技能：每当它被扣减生命值但没有死亡（死亡即生命值 ≤ 0），且 Boss 的随从数量小于上限 k，便会召唤一个新的具有 m 点生命值的“恐怖的奴隶主”。
- 现在小 Y 可以进行 n 次攻击，每次攻击时，会从 Boss 以及 Boss 的所有随从中的等概率随机选择一个，并扣减 1 点生命值，她想知道进行 n 次攻击后扣减 Boss 的生命值点数的期望。为了避免精度误差，你的答案需要对 998244353 取模。
- T 次询问，每次 m 和 k 不变。

例题 1：集训队集训 2017 Day2t3

- 打 Boss，Boss 有 10^{100} 点生命值，但它只带了一个随从——一个只有 m 点生命值的“恐怖的奴隶主”。
- 这个“恐怖的奴隶主”有一个特殊的技能：每当它被扣减生命值但没有死亡（死亡即生命值 ≤ 0），且 Boss 的随从数量小于上限 k，便会召唤一个新的具有 m 点生命值的“恐怖的奴隶主”。
- 现在小 Y 可以进行 n 次攻击，每次攻击时，会从 Boss 以及 Boss 的所有随从中的等概率随机选择一个，并扣减 1 点生命值，她想知道进行 n 次攻击后扣减 Boss 的生命值点数的期望。为了避免精度误差，你的答案需要对 998244353 取模。
- T 次询问，每次 m 和 k 不变。
- $1 \leq T \leq 1000, 1 \leq n \leq 10^{18}, 1 \leq m \leq 3, 1 \leq k \leq 8$。

例题 1：集训队集训 2017 Day2t3

- 记一下每个生命值的随从有多少个，和不能超过 k，只有 $w = 165$ 种状态。

例题 1：集训队集训 2017 Day2t3

- 记一下每个生命值的随从有多少个，和不能超过 k，只有 $w=165$ 种状态。
- 我们要算的其实就是个矩阵乘法而已 $A^n x$

例题 1：集训队集训 2017 Day2t3

- 记一下每个生命值的随从有多少个，和不能超过 k，只有 $w = 165$ 种状态。
- 我们要算的其实就是个矩阵乘法而已 $A^n x$
- 先把 A^{2^j} 倍增好，这部分复杂度 $O(w^3 logn)$

例题 1：集训队集训 2017 Day2t3

- 记一下每个生命值的随从有多少个，和不能超过 k，只有 $w=165$ 种状态。
- 我们要算的其实就是个矩阵乘法而已 $A^n x$
- 先把 A^{2^j} 倍增好，这部分复杂度 $O(w^3 logn)$
- 然后每次要算的其实都只是矩阵乘向量，这部分复杂度 $O(Tw^2 logn)$

例题 1：集训队集训 2017 Day2t3 a better solution?

- 假设我们知道了他的特征多项式 $f(x)$(w 次，首项为 1)

例题 1：集训队集训 2017 Day2t3 a better solution?

- 假设我们知道了他的特征多项式 $f(x)$(w 次，首项为 1)
- 算一下 $x^n \mod f(x)$ 的系数，然后组合一下，最好复杂度是 $O(Twlogwlogn)$

例题 1：集训队集训 2017 Day2t3 a better solution?

- 假设我们知道了他的特征多项式 $f(x)$(w 次，首项为 1)
- 算一下 $x^n \mod f(x)$ 的系数，然后组合一下，最好复杂度是 $O(Twlogwlogn)$
- 计算特征多项式？很不幸需要带 w 个值进去求值，然后再差值，复杂度 $O(w^4)$

例题 1：集训队集训 2017 Day2t3 a better solution?

- 假设我们知道了他的特征多项式 $f(x)$(w 次，首项为 1)
- 算一下 $x^n \mod f(x)$ 的系数，然后组合一下，最好复杂度是 $O(Twlogwlogn)$
- 计算特征多项式？很不幸需要带 w 个值进去求值，然后再差值，复杂度 $O(w^4)$
- 但是可以对 18 种 m, k 求完先打好表。

拟阵

- 我们用独立集定义拟阵

拟阵

- 我们用独立集定义拟阵
- 1. 空集是独立集 = 独立集不是空集

拟阵

- 我们用独立集定义拟阵
- 1. 空集是独立集 = 独立集不是空集
- 2. hereditary property: 独立集的子集还是独立集

拟阵

- 我们用独立集定义拟阵
- 1. 空集是独立集 = 独立集不是空集
- 2. hereditary property: 独立集的子集还是独立集
- 3. augumentation property: 一个比较大的独立集一定有一个元素能加到比较小的独立集里

拟阵

- 我们用独立集定义拟阵
- 1. 空集是独立集 = 独立集不是空集
- 2. hereditary property: 独立集的子集还是独立集
- 3. augumentation property: 一个比较大的独立集一定有一个元素能加到比较小的独立集里
- 最大权独立集

它的几个例子

- 生成森林（图论）

它的几个例子

- 生成森林（图论）
- 线性相关

例题：CTSC2015 App

- 每个任务 i 有 ddl:t_i 以及收益 p_i，如果不晚于 t_i 完成，得到 p_i 收益

例题：CTSC2015 App

- 每个任务 i 有 ddl:t_i 以及收益 p_i，如果不晚于 t_i 完成，得到 p_i 收益
- 每个任务占用一个单位的时间，每次操作之后，求最大收益

例题：CTSC2015 App

- 每个任务 i 有 ddl:t_i 以及收益 p_i，如果不晚于 t_i 完成，得到 p_i 收益
- 每个任务占用一个单位的时间，每次操作之后，求最大收益
- 支持操作 1：ADD t p：表示新添一个截止日期为 t，收益为 p 的任务。

例题：CTSC2015 App

- 每个任务 i 有 ddl:t_i 以及收益 p_i，如果不晚于 t_i 完成，得到 p_i 收益
- 每个任务占用一个单位的时间，每次操作之后，求最大收益
- 支持操作 1：ADD t p：表示新添一个截止日期为 t，收益为 p 的任务。
- 支持操作 2：DEL t p：表示删除一个截止日期为 t，收益为 p 的任务。

例题：CTSC2015 App

- 这也是一个拟阵的例子

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。
- (显然因为他的任务完成时间里至少有一个时间更少的那个是空闲的)

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。
- (显然因为他的任务完成时间里至少有一个时间更少的那个是空闲的)
- 这个 trival 的性质就保证了他是个拟阵，也保证了我们贪心的正确性

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。
- (显然因为他的任务完成时间里至少有一个时间更少的那个是空闲的)
- 这个 trival 的性质就保证了他是个拟阵，也保证了我们贪心的正确性
- 首先加和删我们可以变 dfs 时间线段树变成只有加和撤销的情况

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。
- (显然因为他的任务完成时间里至少有一个时间更少的那个是空闲的)
- 这个 trival 的性质就保证了他是个拟阵，也保证了我们贪心的正确性
- 首先加和删我们可以变 dfs 时间线段树变成只有加和撤销的情况
- 合法的条件就是 $s_i \leq i$，每次加一个之后，第一个加爆的 s_i 前面要踢掉一个。

例题：CTSC2015 App

- 这也是一个拟阵的例子
- 一个更多任务的方案，一定可以丢一个任务给更少任务的方案。
- （显然因为他的任务完成时间里至少有一个时间更少的那个是空闲的）
- 这个 trival 的性质就保证了他是个拟阵，也保证了我们贪心的正确性
- 首先加和删我们可以变 dfs 时间线段树变成只有加和撤销的情况
- 合法的条件就是 $s_i \leq i$，每次加一个之后，第一个加爆的 s_i 前面要踢掉一个。
- 当然是踢掉最小的一个啦，线段树维护一下就好了，撤销可以暴力。

一道趣题

- 一个筛子有 k 个面，不断扔直到大于等于 n。

一道趣题

- 一个筛子有 k 个面，不断扔直到大于等于 n。
- 正好为 n 的概率，在 n 趋向于无穷的情况下收敛到哪?

这和图论有什么关系？

- 有一个 n 个点的折线构成的山脉，两个人鹊巢相会，他们深爱着彼此，所以始终会保持着同一高度。

这和图论有什么关系?

- 有一个 n 个点的折线构成的山脉，两个人鹊巢相会，他们深爱着彼此，所以始终会保持着同一高度。
- 请你输出他们能不能相会。

这和图论有什么关系?

- 有一个 n 个点的折线构成的山脉，两个人鹊巢相会，他们深爱着彼此，所以始终会保持着同一高度。
- 请你输出他们能不能相会。
- $n \leq 10^7$

这和图论有什么关系？

- 先把高度全部离散化

这和图论有什么关系？

- 先把高度全部离散化
- 每个点对表示一个两个人的状态（x, y）

这和图论有什么关系？

- 先把高度全部离散化
- 每个点对表示一个两个人的状态（x, y）
- 每个点对度都为偶数

这和图论有什么关系?

- 先把高度全部离散化
- 每个点对表示一个两个人的状态(x, y)
- 每个点对度都为偶数
- 偏偏起始状态度为奇数?

这和图论有什么关系？

- 先把高度全部离散化
- 每个点对表示一个两个人的状态（x, y）
- 每个点对度都为偶数
- 偏偏起始状态度为奇数？
- 终止状态度数也为奇数！一定可以走到！

集训队集训 day1 t2

- $nm \le 1000$ 的网格图，里面有一些水管。

集训队集训 day1 t2

- $nm \leq 1000$ 的网格图，里面有一些水管。
- 水管可以有很多头（2 到 4 个），其中直线的水管不可以转，别的水管都可以以 1 的代价旋转九十度，可以顺时针或者逆时针转。

集训队集训 day1 t2

- $nm \leq 1000$ 的网格图，里面有一些水管。
- 水管可以有很多头（2 到 4 个），其中直线的水管不可以转，别的水管都可以以 1 的代价旋转九十度，可以顺时针或者逆时针转。
- 请问最少多少词旋转使得没有漏水?

集训队集训 day1 t2

- 比较明显的匹配模型

集训队集训 day1 t2

- 比较明显的匹配模型
- 首先黑白染色分出 S 集和 T 集

集训队集训 day1 t2

- 比较明显的匹配模型
- 首先黑白染色分出 S 集和 T 集
- 然后分类讨论

集训队集训 day1 t2

- 如果是两个头的 L 形，那么限制匹配的一定是一个行点和一个列点

集训队集训 day1 t2

- 如果是两个头的 L 形，那么限制匹配的一定是一个行点和一个列点
- 然后行点和列点变化代价分别为 1

集训队集训 day1 t2

- 如果是两个头的 L 形，那么限制匹配的一定是一个行点和一个列点
- 然后行点和列点变化代价分别为 1
- 如果是三个头就更简单了，给没选的那个带一个负数权值，表示没选有个代价

集训队集训 day1 t2

- 如果是两个头的 L 形，那么限制匹配的一定是一个行点和一个列点
- 然后行点和列点变化代价分别为 1
- 如果是三个头就更简单了，给没选的那个带一个负数权值，表示没选有个代价
- 加一个常数把它变成正的。

POJ chessboard

- $n*n$ 的棋盘上摆了 $n \leq 10^5$ 个车，让他们两两不攻击

POJ chessboard

- $n*n$ 的棋盘上摆了 $n \leq 10^5$ 个车，让他们两两不攻击
- 第 i 个车必须摆在一个矩形里

POJ chessboard

- $n*n$ 的棋盘上摆了 $n \le 10^5$ 个车，让他们两两不攻击
- 第 i 个车必须摆在一个矩形里
- 求哪些车的摆放位置是确定的?

POJ chessboard

- 行列独立

POJ chessboard

- 行列独立
- 二分图匹配

POJ chessboard

- 行列独立
- 二分图匹配
- 每个点匹配是否唯一?

POJ chessboard

- 行列独立
- 二分图匹配
- 每个点匹配是否唯一?
- 往右只保留匹配边，往左只保留未匹配边，它是不是在一个环中

POJ chessboard

- 行列独立
- 二分图匹配
- 每个点匹配是否唯一?
- 往右只保留匹配边，往左只保留未匹配边，它是不是在一个环中
- 缩强连通分量即可

USACO OPEN17 grass

- 动态修改点的颜色，求不同色点间边权的最小值

USACO OPEN17 grass

- 动态修改点的颜色，求不同色点间边权的最小值
- $n \leq 5 * 10^5$

USACO OPEN17 grass

- 动态修改点的颜色，求不同色点间边权的最小值
- $n \le 5 * 10^5$
- 两点间颜色不同，那么任何一个路径上都一定会有不同

USACO OPEN17 grass

- 动态修改点的颜色，求不同色点间边权的最小值
- $n \leq 5 * 10^5$
- 两点间颜色不同，那么任何一个路径上都一定会有不同
- 取最小生成树，每个点维护下孩子的所有颜色

四十五度仰望天空

- 曼哈顿距离

四十五度仰望天空

- 曼哈顿距离
- 切比雪夫距离

四十五度仰望天空

- 曼哈顿距离
- 切比雪夫距离
- 旋转

曼哈顿距离最大生成树

- $n \leq 10^5$

曼哈顿距离最大生成树

- $n \leq 10^5$
- $n \leq 10^6$

曼哈顿距离最大生成树

- $n \leq 10^5$
- $n \leq 10^6$
- easy

bzoj4061

- 一个 $n \leq 10^5$ 个点 $m \leq 3 * 10^5$ 个边的无向图

bzoj4061

- 一个 $n \leq 10^5$ 个点 $m \leq 3 * 10^5$ 个边的无向图
- 新建一个点，确定它到每个边的路径长度，非负可以不是整数

bzoj4061

- 一个 $n \le 10^5$ 个点 $m \le 3 * 10^5$ 个边的无向图
- 新建一个点，确定它到每个边的路径长度，非负可以不是整数
- 使得每个点到 1 和 2 的最短路都不经过他

bzoj4061

- 一个 $n \le 10^5$ 个点 $m \le 3 * 10^5$ 个边的无向图
- 新建一个点，确定它到每个边的路径长度，非负可以不是整数
- 使得每个点到 1 和 2 的最短路都不经过他
- 最小化它到每个点的距离的和

bzoj4061

- 先求出每个点到 1 和 2 的最短路

bzoj4061

- 先求出每个点到 1 和 2 的最短路
- x_i 表示这个点到 i 的距离，可以列出方程 $x_1 + x_i \geq k$

bzoj4061

- 先求出每个点到 1 和 2 的最短路
- x_i 表示这个点到 i 的距离，可以列出方程 $x_1 + x_i \geq k$
- 显然在满足方程的情况下每个点的距离 x_i 越小越好

bzoj4061

- 先求出每个点到 1 和 2 的最短路
- x_i 表示这个点到 i 的距离，可以列出方程 $x_1 + x_i \geq k$
- 显然在满足方程的情况下每个点的距离 x_i 越小越好
- $x_i = max(|x_1 - k_1|, |x_2 - k_2|)$

bzoj4061

- 先求出每个点到 1 和 2 的最短路
- x_i 表示这个点到 i 的距离，可以列出方程 $x_1 + x_i \geq k$
- 显然在满足方程的情况下每个点的距离 x_i 越小越好
- $x_i = max(|x_1 - k_1|, |x_2 - k_2|)$
- 最小化 $\sum x_i$，这是个切比雪夫距离，转 45 度变成曼哈顿距离

bzoj4061

- 先求出每个点到 1 和 2 的最短路
- x_i 表示这个点到 i 的距离，可以列出方程 $x_1 + x_i \geq k$
- 显然在满足方程的情况下每个点的距离 x_i 越小越好
- $x_i = max(|x_1 - k_1|, |x_2 - k_2|)$
- 最小化 $\sum x_i$，这是个切比雪夫距离，转 45 度变成曼哈顿距离
- 两维取中位数即可

Schwartz–Zippel lemma

- 一个 m 元 n 次多项式，在域 F 内随机给每个变量赋值。

Schwartz–Zippel lemma

- 一个 m 元 n 次多项式，在域 F 内随机给每个变量赋值。
- 等于零的概率小于 $\frac{n}{|F|}$

集训队集训 2014：结合律

- 给你一个运算，请你判断他是否满足结合律?

集训队集训 2014：结合律

- 给你一个运算，请你判断他是否满足结合律?
- $n \leq 5000$

集训队集训 2014：结合律

- std 做法：

集训队集训 2014：结合律

- std 做法：

  ## 1/8 approach

  首先证明一个lemma: 定义S^k的一个minor是它的k个子集的笛卡尔积，如果有一个不等于0的函数$f: S^k->H$（H是一个阿贝尔群），定义minor的f值是他所有元素的f值之和。那么至少有$\frac{1}{2^k}$的minor T都满足$f(T)\neq 0$。

  考虑固定一个不为0的元素 $(a_1,a_2\dots a_k)$，我们把所有minor按照它和$(S/a_1, S/a_2\dots, S/a_k)$的交归类，每类里有$2^k$个minor，他们之间只有第$i$维是否包含$a_i$不同。

  然后我们考虑对每类内用容斥原理表示出(这里用到了H是阿贝尔群的性质)$(a_1,a_2\dots a_k)$，实际上就是t维强制不是a的时候权值为$(-1)^t$。

  然后由于这个元素不为0，所以每组至少有一个minor不为0，从而至少有$\frac{1}{2^k}$的minor不为0.

  考虑现在要做的是$f(S^3)=(ab)c-a(bc)=(ab)c+a(bc), f:S^3->S/2$, 定义$S/2=(F_2)[S]=\{\sigma_i s_i|\sigma_i\in F_2\}$，类似于一个S上的01多项式，我们类似多项式系数mod 2意义下的乘法定义$S/2$上的乘法。

  $(ab)c-a(bc)=(ab)c+a(bc)$是因为我们对系数mod 2了(系数在F_2上)，显然是不会影响是否满足结合律的。

  如果S不满足结合律，那么随机S^3的一个minor，也就是$(S/2)^3$，至少会有$\frac{1}{8}$的概率出错。

  考虑$f(T)=\sum_{t\in T} f(t)$，要判它等于0，实际上等于判$S/2$里满足结合律。

- 我们随机了三个$S/2$里的多项式之后，可以暴力$O(n^2)$多项式乘法验证。

集训队集训 2014：结合律

- 太复杂了

集训队集训 2014：结合律

- 太复杂了
- 考虑 $\mod p$ 随机的时候，每项是个几次多项式

NOI2013

- 判断是否有 $AB = C$?

NOI2013

- 判断是否有 $AB = C$?
- 随机 x

NOI2013

- 判断是否有 $AB = C$?
- 随机 x
- 错误概率? $(AB - C)x = 0$, $modp$ 意义下为 $\frac{p^n}{p^(n-1)} = \frac{1}{p}$

线性规划和对偶原理

- $\max(c^t x | Ax \leq b) = \min(b^t y | A^t y \geq c)$

线性规划和对偶原理

- $\max(c^t x|Ax \le b) = \min(b^t y|A^t y \ge c)$
- 互相松弛定理 $(b - Ax)^t y = 0, (c - A^t y)^t x = 0$

线性规划和对偶原理

- $\max(c^t x | Ax \leq b) = \min(b^t y | A^t y \geq c)$
- 互相松弛定理 $(b - Ax)^t y = 0, (c - A^t y)^t x = 0$
- 组合意义（经济学博弈）

线性规划和对偶原理

- $\max(c^t x | Ax \le b) = \min(b^t y | A^t y \ge c)$
- 互相松弛定理 $(b - Ax)^t y = 0, (c - A^t y)^t x = 0$
- 组合意义（经济学博弈）
- 顺便一提，大家熟悉的斜率优化

对偶原理

- $max\{c^Tx|Ax \le b\} = min\{b^Ty|A^Ty \ge c\}$

对偶原理

- $max\{c^Tx|Ax \leq b\} = min\{b^Ty|A^Ty \geq c\}$
- why? 我们来直观的理解一下

对偶原理

- $max\{c^Tx|Ax \le b\} = min\{b^Ty|A^Ty \ge c\}$
- why？我们来直观的理解一下
- 假设你是工厂主，每个产品的收益为 c, 每个产品需要的材料是 A，你有的原材料个数是 b

对偶原理

- $max\{c^T x | Ax \leq b\} = min\{b^T y | A^T y \geq c\}$
- why？我们来直观的理解一下
- 假设你是工厂主，每个产品的收益为 c, 每个产品需要的材料是 A，你有的原材料个数是 b
- 决策生产个数 x 最大化收益（忽略加工费之类的）

对偶原理

- $max\{c^Tx|Ax \le b\} = min\{b^Ty|A^Ty \ge c\}$
- why? 我们来直观的理解一下
- 假设你是工厂主，每个产品的收益为 c, 每个产品需要的材料是 A，你有的原材料个数是 b
- 决策生产个数 x 最大化收益（忽略加工费之类的）
- 反过来你要卖给工厂主这些材料，决策原材料单价 y，你至少卖给他多贵才不会亏

对偶原理

- $max\{c^Tx|Ax \leq b\} = min\{b^Ty|A^Ty \geq c\}$
- why？我们来直观的理解一下
- 假设你是工厂主，每个产品的收益为 c, 每个产品需要的材料是 A，你有的原材料个数是 b
- 决策生产个数 x 最大化收益（忽略加工费之类的）
- 反过来你要卖给工厂主这些材料，决策原材料单价 y，你至少卖给他多贵才不会亏
- y 称为『影子价格』是对价格的估计，显然因为忽略了工厂主的社会必要劳动时间，所以这两个问题的答案应该是相等的（成本 = 收益）

SRM676 L3

- 有一个技能 DAG，你要点满所有技能点，每个技能点需要时间 t

SRM676 L3

- 有一个技能 DAG，你要点满所有技能点，每个技能点需要时间 t
- 你有 C 元，每次可以氪金 c_i 元来给一个技能 $-1s$，求你点完所有技能点的最小时间（可以同时点多个技能点，但是这多个之间不能有依赖关系）

SRM676 L3

- 有一个技能 DAG，你要点满所有技能点，每个技能点需要时间 t
- 你有 C 元，每次可以氪金 c_i 元来给一个技能 $-1s$，求你点完所有技能点的最小时间（可以同时点多个技能点，但是这多个之间不能有依赖关系）
- $n \leq 50$

SRM676 L3

- 二分答案 mid

SRM676 L3

- 二分答案 mid
- 对偶之后变成了选最若干个路径，每个边至多选 c_i 次，最大化每条路径 $-mid$ 的权值和

SRM676 L3

- 二分答案 mid
- 对偶之后变成了选最若干个路径，每个边至多选 c_i 次，最大化每条路径 $-mid$ 的权值和
- DAG 上的费用流问题

SRM676 L3

- 二分答案 mid
- 对偶之后变成了选最若干个路径，每个边至多选 c_i 次，最大化每条路径 $-mid$ 的权值和
- DAG 上的费用流问题
- 当然可以拆点最小割，但是没有对偶优雅简洁。

cf352D

- 一个有 n 个点的树，树上的边已经全部损坏了。现在有 m 个工人，每个工人可以修复从 u_i 到 v_i (保证 v_i 是 u_i 到 $root$ 上的一个点) 上的所有边，花费为 c_i。问把所有边修好的最小花费为多少?

cf352D

- 一个有 n 个点的树，树上的边已经全部损坏了。现在有 m 个工人，每个工人可以修复从 u_i 到 v_i (保证 v_i 是 u_i 到 $root$ 上的一个点) 上的所有边，花费为 c_i。问把所有边修好的最小花费为多少?
- $n \leq 3 * 10^5$

cf352D

- 一个有 n 个点的树，树上的边已经全部损坏了。现在有 m 个工人，每个工人可以修复从 u_i 到 v_i (保证 v_i 是 u_i 到 $root$ 上的一个点) 上的所有边，花费为 c_i。问把所有边修好的最小花费为多少?
- $n \leq 3 * 10^5$
- Feel the magic of dual!

cf352D

- 传统做法大概 dp 一下

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价
- 相信大家在 OI 中已经对这样的做法审美疲劳了

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价
- 相信大家在 OI 中已经对这样的做法审美疲劳了
- 对偶做法: 考虑对偶问题是『选最多的边，使得每个工人可以修复的范围内少于 c_i 个』

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价
- 相信大家在 OI 中已经对这样的做法审美疲劳了
- 对偶做法: 考虑对偶问题是『选最多的边，使得每个工人可以修复的范围内少于 c_i 个』
- magical greedy!

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价
- 相信大家在 OI 中已经对这样的做法审美疲劳了
- 对偶做法: 考虑对偶问题是『选最多的边，使得每个工人可以修复的范围内少于 c_i 个』
- magical greedy!
- 可以选的时候一定是选了更优

cf352D

- 传统做法大概 dp 一下
- f_i 表示完成 fa_i 到 i 的边和子树的最小代价，转移在回溯的时候维护每个末端点的代价
- 相信大家在 OI 中已经对这样的做法审美疲劳了
- 对偶做法: 考虑对偶问题是『选最多的边，使得每个工人可以修复的范围内少于 c_i 个』
- magical greedy!
- 可以选的时候一定是选了更优
- 判断可不可以选可以 set 维护一下剩余多少个，启发式合并

互补松弛定理

- 互补松弛定理

互补松弛定理

- 互补松弛定理
- 让我们回到工厂，现在如果一个东西是亏的（右边不等式没有取到等号），那么你还会去买吗?

互补松弛定理

- 互补松弛定理
- 让我们回到工厂，现在如果一个东西是亏的（右边不等式没有取到等号)，那么你还会去买吗?
- 对偶问题里不等式取不到等号，x_i 为 0。逆否命题，x_i 大于 0，一定取等号。

互补松弛定理

- 互补松弛定理
- 让我们回到工厂，现在如果一个东西是亏的（右边不等式没有取到等号），那么你还会去买吗?
- 对偶问题里不等式取不到等号，x_i 为 0。逆否命题，x_i 大于 0，一定取等号。
- 我们现在是知道了 y 求 x，对偶一下

互补松弛定理

- 互补松弛定理
- 让我们回到工厂，现在如果一个东西是亏的（右边不等式没有取到等号），那么你还会去买吗？
- 对偶问题里不等式取不到等号，x_i 为 0。逆否命题，x_i 大于 0，一定取等号。
- 我们现在是知道了 y 求 x，对偶一下
- 影子价格不为 0 当且仅当对应的对偶问题不等式取等号

互补松弛定理

- 互补松弛定理
- 让我们回到工厂，现在如果一个东西是亏的（右边不等式没有取到等号），那么你还会去买吗?
- 对偶问题里不等式取不到等号，x_i 为 0。逆否命题，x_i 大于 0，一定取等号。
- 我们现在是知道了 y 求 x，对偶一下
- 影子价格不为 0 当且仅当对应的对偶问题不等式取等号
- $x^T(c - A^Ty) = 0, y^T(b - Ax) = 0$，高斯消元

例题：Delight for a cat

- 有一只喵，共 n 天，它每 k 天吃吃吃的天数要在 $[l, r]$ 之间，每天睡觉有个收益 s_i，吃猫粮有个收益 e_i，求最大收益。

例题：Delight for a cat

- 有一只喵，共 n 天，它每 k 天吃吃吃的天数要在 $[l, r]$ 之间，每天睡觉有个收益 s_i，吃猫粮有个收益 e_i，求最大收益。
- $k, n \leq 1000$

例题：Delight for a cat

- 有一只喵，共 n 天，它每 k 天吃吃吃的天数要在 $[l, r]$ 之间，每天睡觉有个收益 s_i，吃猫粮有个收益 e_i，求最大收益。
- $k, n \leq 1000$
- 输出方案

线性规划转网络流

- 解法 1：每个限制的天数都是连续的，可以对偶差分

线性规划转网络流

- 解法 1：每个限制的天数都是连续的，可以对偶差分
- 解法 2：每天在限制上也是连续的，可以直接差分不对偶

纳什均衡

- 纳什均衡

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）
- 混合策略下的纳什均衡（一定存在）

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）
- 混合策略下的纳什均衡（一定存在）
- 零和游戏

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）
- 混合策略下的纳什均衡（一定存在）
- 零和游戏
- 零和游戏固定策略下的后手优势 $(min_i max_j A_i, j \geq max_j min_i A_i, j)$

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）
- 混合策略下的纳什均衡（一定存在）
- 零和游戏
- 零和游戏固定策略下的后手优势 ($min_i max_j A_i, j \geq max_j min_i A_i, j$)
- 零和游戏混合策略纳什均衡，与对偶原理

纳什均衡

- 纳什均衡
- 固定策略纳什均衡（不总是存在——剪刀石头布）
- 混合策略下的纳什均衡（一定存在）
- 零和游戏
- 零和游戏固定策略下的后手优势 $(min_i max_j A_i, j \geq max_j min_i A_i, j)$
- 零和游戏混合策略纳什均衡，与对偶原理
- Yao's Principle

例题

- $n \times n$ 的网格图，你要给每个点放一些需要 $f_{i,j} \in R$ 时间通过的路障，要求 $\forall i \sum_{j=1}^{n} f_{i,j} = 1$。

例题

- $n \times n$ 的网格图，你要给每个点放一些需要 $f_{i,j} \in R$ 时间通过的路障，要求 $\forall i \sum_{j=1}^{n} f_{i,j} = 1$。
- 现在从 $(0,0)$ 点出发，每次可以几乎不用时间的从 (i,j) 走到 $(i+1, j-1), (i+1, j), (i+1, j+1)$

例题

- $n \times n$ 的网格图，你要给每个点放一些需要 $f_{i,j} \in R$ 时间通过的路障，要求 $\forall i \sum_{j=1}^{n} f_{i,j} = 1$。
- 现在从 $(0,0)$ 点出发，每次可以几乎不用时间的从 (i,j) 走到 $(i+1,j-1),(i+1,j),(i+1,j+1)$
- 求最短路径最长是多少?

例题

- $n \times n$ 的网格图，你要给每个点放一些需要 $f_{i,j} \in R$ 时间通过的路障，要求 $\forall i \sum_{j=1}^{n} f_{i,j} = 1$。
- 现在从 $(0,0)$ 点出发，每次可以几乎不用时间的从 (i,j) 走到 $(i+1, j-1), (i+1, j), (i+1, j+1)$
- 求最短路径最长是多少?
- $n \leq 10^{1}0$，保留 3 位小数。

例题

- $ans = H_n$。

例题

- $ans = H_n$。
- 随机策略