Making Graphs into Trees

make_pair(immortalCO,WrongAnswer)

2017 年 3 月 11 日

- 在 OI 中，有一些图的问题，如果能够将图转化成树，可能有一些很方便的解法

- 在 OI 中，有一些图的问题，如果能够将图转化成树，可能有一些很方便的解法
- 我们总结了一些将图转化为树的方法：DFS 树、BFS 树和圆方树。

DFS 树

- 图 $G = (V, E)$ 的 DFS 树 $T = (V, E')$ 是对图 G 进行 DFS 的过程中，经过的 E 中的边构成的树。

DFS 树

- 图 $G=(V,E)$ 的 DFS 树 $T=(V,E')$ 是对图 G 进行 DFS 的过程中，经过的 E 中的边构成的树。
- 经过的边 $e\in E'$ 称为树边，不经过的边 $e\in E-E'$ 称为非树边。

DFS 树

- 图 $G=(V,E)$ 的 DFS 树 $T=(V,E')$ 是对图 G 进行 DFS 的过程中，经过的 E 中的边构成的树。
- 经过的边 $e\in E'$ 称为树边，不经过的边 $e\in E-E'$ 称为非树边。
- DFS 需要先选择一个点 $r\in V$ 作为 T 的根。

DFS 树的性质

1. 任意非树边 $(u, v) \in E - E'$，满足在 T 中 u 是 v 的祖先或 v 是 u 的祖先。

DFS 树的性质

1. 任意非树边 $(u, v) \in E - E'$，满足在 T 中 u 是 v 的祖先或 v 是 u 的祖先。
2. 对于非树边 (u, v)，我们称树 T 中 u 和 v 之间的边被 (u, v) 覆盖。

用 DFS 树处理仙人掌

- 首先说明一下，由于本人是个暴力选手，做仙人掌题最多也只会打打暴力。

用 DFS 树处理仙人掌

- 首先说明一下，由于本人是个暴力选手，做仙人掌题最多也只会打打暴力。
- 找仙人掌的环，只要 DFS 得到一个 DFS 树，那么每条非树边对应一个环。

用 DFS 树处理仙人掌

- 首先说明一下，由于本人是个暴力选手，做仙人掌题最多也只会打打暴力。
- 找仙人掌的环，只要 DFS 得到一个 DFS 树，那么每条非树边对应一个环。
- 一个环就是一条非树边 $e \in E - E'$ 以及 e 覆盖的所有树边。由于任意两个环的边不相交，所以任意两条非树边 $e_1, e_2 \in E - E'$，$e_1 \neq e_2$，有 e_1 和 e_2 覆盖的树边集合不相交。

用 DFS 树处理仙人掌

- 首先说明一下，由于本人是个暴力选手，做仙人掌题最多也只会打打暴力。
- 找仙人掌的环，只要 DFS 得到一个 DFS 树，那么每条非树边对应一个环。
- 一个环就是一条非树边 $e \in E - E'$ 以及 e 覆盖的所有树边。由于任意两个环的边不相交，所以任意两条非树边 $e_1, e_2 \in E - E'$，$e_1 \neq e_2$，有 e_1 和 e_2 覆盖的树边集合不相交。
- 这样就能很方便找出每一条边属于哪一个环。

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法：

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法：
- 首先对仙人掌 DFS 得到 DFS 树，然后求这个树的独立集，$f(i, j)$ 表示当 i 的父节点状态为 j 时的最大独立集。但是——

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法：
- 首先对仙人掌 DFS 得到 DFS 树，然后求这个树的独立集，$f(i,j)$ 表示当 i 的父节点状态为 j 时的最大独立集。但是——
- 树的独立集不一定是仙人掌的独立集，因为不能满足非树边的约束。

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法：
- 首先对仙人掌 DFS 得到 DFS 树，然后求这个树的独立集，$f(i,j)$ 表示当 i 的父节点状态为 j 时的最大独立集。但是——
- 树的独立集不一定是仙人掌的独立集，因为不能满足非树边的约束。
- 怎么办?

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法:
- 首先对仙人掌 DFS 得到 DFS 树，然后求这个树的独立集，$f(i,j)$ 表示当 i 的父节点状态为 j 时的最大独立集。但是——
- 树的独立集不一定是仙人掌的独立集，因为不能满足非树边的约束。
- 怎么办?
- 在状态里加一维表示 i 所在环的顶端（环中深度最小的点）的状态 k，即记 $f(i,j,k)$ 为 i 的父节点状态为 j 且 i 所在环顶状态为 k 时，i 为根的子树的最大独立集。

仙人掌独立集

- 给定仙人掌，求仙人掌的最大独立集。
- 这个问题 CJK 给出了非常优秀的解法，然而我在做这道题的时候什么也不会，就想了一个做法：
- 首先对仙人掌 DFS 得到 DFS 树，然后求这个树的独立集，$f(i,j)$ 表示当 i 的父节点状态为 j 时的最大独立集。但是——
- 树的独立集不一定是仙人掌的独立集，因为不能满足非树边的约束。
- 怎么办?
- 在状态里加一维表示 i 所在环的顶端（环中深度最小的点）的状态 k，即记 $f(i,j,k)$ 为 i 的父节点状态为 j 且 i 所在环顶状态为 k 时，i 为根的子树的最大独立集。
- 当 i 在环底端且 $k=1$ 时不能选 i 即可。于是就做完了。复杂度 $O(n)$，n 可以理解为 $|V|$ 或者 $|E|$（两者同阶）。

用 DFS 树进行状压树形 DP

- 受到之前 DFS 树性质的启发，我们可以用它来解决一些稍微复杂的问题。

用 DFS 树进行状压树形 DP

- 受到之前 DFS 树性质的启发，我们可以用它来解决一些稍微复杂的问题。
- “状压树形 DP”?

图染色方案计数

- 给定无向连通图 $G = (V, E)$，求用 k 种颜色对 V 染色，且对于任意 $(u, v) \in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V| \le 100, |E| < |V| + 10$。

图染色方案计数

- 给定无向连通图 $G = (V, E)$，求用 k 种颜色对 V 染色，且对于任意 $(u, v) \in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V| \leq 100, |E| < |V| + 10$。
- 图染色问题？NPC 问题？

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？显然可以树形 DP，记 $f(i)$ 为以 i 为根的子树有多少种染色方案，如果 i 是根，那么 i 有 k 种，否则 i 有 $k-1$ 种，然后再乘上 i 的所有子节点 j 的 $f(j)$ 值。

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
$|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？显然可以树形 DP，记 $f(i)$ 为以 i 为根的子树有多少种染色方案，如果 i 是根，那么 i 有 k 种，否则 i 有 $k-1$ 种，然后再乘上 i 的所有子节点 j 的 $f(j)$ 值。
- 有环怎么搞？

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？显然可以树形 DP，记 $f(i)$ 为以 i 为根的子树有多少种染色方案，如果 i 是根，那么 i 有 k 种，否则 i 有 $k-1$ 种，然后再乘上 i 的所有子节点 j 的 $f(j)$ 值。
- 有环怎么搞？同样的，构造 DFS 树，那么非树边不超过 10 条，在树形 DP 的状态中暴力记录每一对非树边的顶端颜色是否相同。这实际上就是 10 个元素的集合划分，用最小表示法来记录，不同的划分只有 $Bell(10)=115975$ 种。

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？显然可以树形 DP，记 $f(i)$ 为以 i 为根的子树有多少种染色方案，如果 i 是根，那么 i 有 k 种，否则 i 有 $k-1$ 种，然后再乘上 i 的所有子节点 j 的 $f(j)$ 值。
- 有环怎么搞？同样的，构造 DFS 树，那么非树边不超过 10 条，在树形 DP 的状态中暴力记录每一对非树边的顶端颜色是否相同。这实际上就是 10 个元素的集合划分，用最小表示法来记录，不同的划分只有 $Bell(10)=115975$ 种。
- 记 $n=|V|, m=|E|$，复杂度 $O(nB_{m-n+1})$。事实上有一种更好实现的做法：可以直接外面枚举 10 个点的颜色划分，里面做树形 DP。

图染色方案计数

- 给定无向连通图 $G=(V,E)$，求用 k 种颜色对 V 染色，且对于任意 $(u,v)\in E$，u 和 v 的颜色不同的方案数，答案模 p。
  $|V|\le 100, |E|<|V|+10$。
- 图染色问题？NPC 问题？
- 如果 G 是树怎么做？显然可以树形 DP，记 $f(i)$ 为以 i 为根的子树有多少种染色方案，如果 i 是根，那么 i 有 k 种，否则 i 有 $k-1$ 种，然后再乘上 i 的所有子节点 j 的 $f(j)$ 值。
- 有环怎么搞？同样的，构造 DFS 树，那么非树边不超过 10 条，在树形 DP 的状态中暴力记录每一对非树边的顶端颜色是否相同。这实际上就是 10 个元素的集合划分，用最小表示法来记录，不同的划分只有 $Bell(10)=115975$ 种。
- 记 $n=|V|, m=|E|$，复杂度 $O(nB_{m-n+1})$。事实上有一种更好实现的做法：可以直接外面枚举 10 个点的颜色划分，里面做树形 DP。
- 给大家讲个笑话：CJK 把这题出到了 NOIP 校内训练。

优化复杂度

- 注意到如果是树，可以不用真正跑一遍树形 DP，记 $n = |V|$，则答案就是 $k(k-1)^{n-1}$。因此对于图，复杂度也是有优化空间的。

优化复杂度

- 注意到如果是树，可以不用真正跑一遍树形 DP，记 $n = |V|$，则答案就是 $k(k-1)^{n-1}$。因此对于图，复杂度也是有优化空间的。
- 当 $n > 1$ 时，每个叶节点会使答案乘 $k-1$，所以可以不断删掉叶节点；之后每个点度数至少为 2，再考虑度数为 2 的点，把度数为 2 的点构成的链缩成一条边。缩完以后每个染色方案在原图中对应了多种染色方案，我们可以 DP 预处理一条链两端相同和两端不同时链上的染色方案数 a_i, b_i，那么对于一条长度为 l 的链，缩成一条边以后，边的两个权值为 a_l 和 b_l。那么答案就是所有染色方案的所有边权（两端颜色相同则取前者，否则取后者）乘积之和。把状态转移方程改一改就行了。

优化复杂度

- 注意到如果是树，可以不用真正跑一遍树形 DP，记 $n = |V|$，则答案就是 $k(k-1)^{n-1}$。因此对于图，复杂度也是有优化空间的。
- 当 $n > 1$ 时，每个叶节点会使答案乘 $k-1$，所以可以不断删掉叶节点；之后每个点度数至少为 2，再考虑度数为 2 的点，把度数为 2 的点构成的链缩成一条边。缩完以后每个染色方案在原图中对应了多种染色方案，我们可以 DP 预处理一条链两端相同和两端不同时链上的染色方案数 a_i, b_i，那么对于一条长度为 l 的链，缩成一条边以后，边的两个权值为 a_l 和 b_l。那么答案就是所有染色方案的所有边权（两端颜色相同则取前者，否则取后者）乘积之和。把状态转移方程改一改就行了。
- 这样图满足 $\forall v \in V, \deg(v) \geq 3$，因为 $\sum_{v\in V}\deg(v) = 2m \geq 3n$，所以 $n \leq 2(m-n)$，因此复杂度优化为 $O((m-n)B_{m-n+1})$。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。
- 给定无向图 $G = (V, E)$，求 G 的大小为 k 的独立集个数，答案对 p 取模。$|E| - |V|$ 很小。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。
- 给定无向图 $G = (V, E)$，求 G 的大小为 k 的独立集个数，答案对 p 取模。$|E| - |V|$ 很小。
- 受到之前的启发，这个问题已经很容易解决了。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。
- 给定无向图 $G=(V,E)$，求 G 的大小为 k 的独立集个数，答案对 p 取模。$|E|-|V|$ 很小。
- 受到之前的启发，这个问题已经很容易解决了。
- 求一个 DFS 树 $T=(V,E)$。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。
- 给定无向图 $G=(V,E)$，求 G 的大小为 k 的独立集个数，答案对 p 取模。$|E|-|V|$ 很小。
- 受到之前的启发，这个问题已经很容易解决了。
- 求一个 DFS 树 $T=(V,E)$。
- 记 $f(i,j,S)$ 为 T 中以 i 为根的子树中大小为 j 的独立集数。S 是做什么用的？S 里面是一些约束条件，限制了某些点不能选。注意到覆盖 i 和 i 父节点之间的连边的非树边集合 C_i 不大，可以在 S 中记下 i 的父节点以及 C_i 中所有边的顶端是否被选。

改编：大小为 k 的独立集计数

- 这是我对上一题的改编。
- 给定无向图 $G = (V, E)$，求 G 的大小为 k 的独立集个数，答案对 p 取模。$|E| - |V|$ 很小。
- 受到之前的启发，这个问题已经很容易解决了。
- 求一个 DFS 树 $T = (V, E)$。
- 记 $f(i, j, S)$ 为 T 中以 i 为根的子树中大小为 j 的独立集数。S 是做什么用的？S 里面是一些约束条件，限制了某些点不能选。注意到覆盖 i 和 i 父节点之间的连边的非树边集合 C_i 不大，可以在 S 中记下 i 的父节点以及 C_i 中所有边的顶端是否被选。
- 记 $n = |V|$，复杂度 $O(n^2 \sum_{v \in V} 2^{|C_v|})$。这样表示的好处是：即使 $|E| - |V|$ 很大，如果覆盖每条树边的非树边都不多，这个算法仍然可以很快地求出答案。

DFS 树和 BFS 树

- 讲了这么多 DFS 树，其实我们还有一种类似的树：广度优先搜索（BFS）树。

DFS 树和 BFS 树

- 讲了这么多 DFS 树，其实我们还有一种类似的树：广度优先搜索（BFS）树。
- 这个树的性质并不是很适合树形 DP，因为非树边一定不是指向祖先的，但它有另外一个性质：如果把每个点按 BFS 树中的深度分层，那么每条非树边或者在同一层，或者在相邻层。

DFS 树和 BFS 树

- 讲了这么多 DFS 树，其实我们还有一种类似的树：广度优先搜索（BFS）树。
- 这个树的性质并不是很适合树形 DP，因为非树边一定不是指向祖先的，但它有另外一个性质：如果把每个点按 BFS 树中的深度分层，那么每条非树边或者在同一层，或者在相邻层。
- 这个思想可以用于某些分层图 DP，虽然不是很实用就是了……(因此我并没有准备例题)

推荐一道题

- 大家有兴趣可以做一下这道题：UOJ #259

推荐一道题

- 大家有兴趣可以做一下这道题：UOJ #259
- 这是本人原创的提交答案题，用我今天介绍的东西可以解决不少的测试点。

推荐一道题

- 大家有兴趣可以做一下这道题：UOJ #259
- 这是本人原创的提交答案题，用我今天介绍的东西可以解决不少的测试点。
- 提示：不一定要以 1 号点为根。

——————分割线——————

接下来我们来讲讲圆方树

(图片来自知乎)

圆方树是什么东东?

- 当今 OI 界，仙人掌的题目得到了很大的发展

圆方树是什么东东?

- 当今 OI 界，仙人掌的题目得到了很大的发展
- 各种奇奇怪怪的序列维护的题，都被出到树上甚至仙人掌上

圆方树是什么东东?

- 当今 OI 界，仙人掌的题目得到了很大的发展
- 各种奇奇怪怪的序列维护的题，都被出到树上甚至仙人掌上
- 连虚树、树分治、树剖之类的东西，都可以出到仙人掌上

圆方树是什么东东?

- 当今 OI 界，仙人掌的题目得到了很大的发展
- 各种奇奇怪怪的序列维护的题，都被出到树上甚至仙人掌上
- 连虚树、树分治、树剖之类的东西，都可以出到仙人掌上
- 因此，如果有一种通用的方法可以解决一系列仙人掌的问题，那就是非常资磁的呀

圆方树是什么东东?

- 当今 OI 界，仙人掌的题目得到了很大的发展
- 各种奇奇怪怪的序列维护的题，都被出到树上甚至仙人掌上
- 连虚树、树分治、树剖之类的东西，都可以出到仙人掌上
- 因此，如果有一种通用的方法可以解决一系列仙人掌的问题，那就是非常资磁的呀
- ——而圆方树，就是解决大多数静态仙人掌问题的通用技巧!

来，我们下一个定义！

定义：仙人掌

仙人掌是满足每条边只在不超过 1 个简单环中的无向连通图。

来，我们下一个定义！

定义：仙人掌

仙人掌是满足每条边只在不超过 1 个简单环中的无向连通图。

定义：圆方树

仙人掌 $G = (V, E)$ 的圆方树 $T = (V_T, E_T)$ 为满足以下条件的无向图：

来，我们下一个定义！

定义：仙人掌

仙人掌是满足每条边只在不超过 1 个简单环中的无向连通图。

定义：圆方树

仙人掌 $G=(V,E)$ 的圆方树 $T=(V_T,E_T)$ 为满足以下条件的无向图：

- $V_T=R_T\cup S_T, R_T=V, R_T\cap S_T=\emptyset$，我们称 R_T 集合为**圆点**、S_T 集合为**方点**

来，我们下一个定义！

定义：仙人掌

仙人掌是满足每条边只在不超过 1 个简单环中的无向连通图。

定义：圆方树

仙人掌 $G=(V,E)$ 的圆方树 $T=(V_T,E_T)$ 为满足以下条件的无向图：

- $V_T=R_T\cup S_T, R_T=V, R_T\cap S_T=\emptyset$，我们称 R_T 集合为**圆点**、S_T 集合为**方点**
- $\forall e\in E$，若 e 不在任何简单环中，则 $e\in E_T$

来，我们下一个定义！

定义：仙人掌

仙人掌是满足每条边只在不超过 1 个简单环中的无向连通图。

定义：圆方树

仙人掌 $G=(V,E)$ 的圆方树 $T=(V_T,E_T)$ 为满足以下条件的无向图：

- $V_T=R_T\cup S_T, R_T=V, R_T\cap S_T=\emptyset$，我们称 R_T 集合为**圆点**、S_T 集合为**方点**
- $\forall e\in E$，若 e 不在任何简单环中，则 $e\in E_T$
- 对于每个仙人掌中的简单环 R，存在方点 $p_R\in S_T$，并且 $\forall p\in R$ 满足 $(p_R,p)\in E_T$，即对每个环建方点连所有点

为什么这定义出的是树呢?

- 我们来证明定义的正确性（圆方树是树）

证明.

- 不在环上的边在圆方树中依然存在，因此这些边连通性不变；每个环通过新建方点的方式连成一朵菊花，连通性也不变，因此圆方树是无向连通图

为什么这定义出的是树呢?

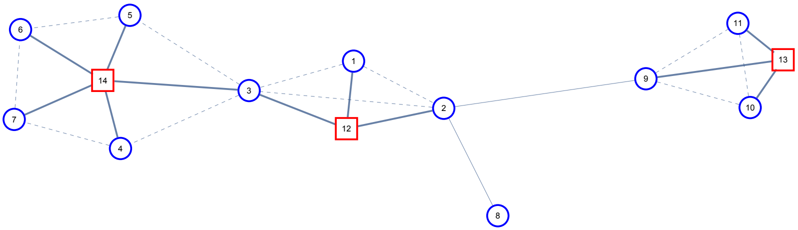
- 我们来证明定义的正确性（圆方树是树）

证明.

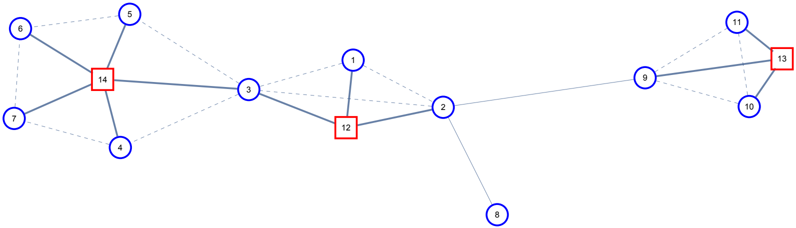
- 不在环上的边在圆方树中依然存在，因此这些边连通性不变；每个环通过新建方点的方式连成一朵菊花，连通性也不变，因此圆方树是无向连通图
- 原图中环的个数为 $|E|-|V|+1$，则 $|V_T|=|S_T|+|R_T|=|V|+|E|-|V|+1=|E|+1$，$|E_T|=|E|$（大小为 r 的环在仙人掌和圆方树中都是 r 条边），因此满足 $|V_T|=|E_T|+1$

□

举个栗子

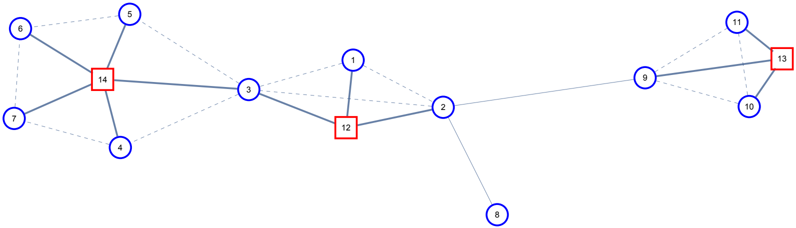


举个栗子



- 是不是很熟悉?

举个栗子



- 是不是很熟悉？这是《火车司机出秦川》的样例一。

如何构造圆方树?

1. 从任意一个个点开始运行 Tarjan 求点双连通分量算法

如何构造圆方树?

1. 从任意一个个点开始运行 Tarjan 求点双连通分量算法
2. 对于每个点双，我们栈中取出，这时栈中的顺序就是环上的顺序，在圆方树中建立方点，依次向栈中的圆点连边

如何构造圆方树?

1. 从任意一个个点开始运行 Tarjan 求点双连通分量算法
2. 对于每个点双，我们栈中取出，这时栈中的顺序就是环上的顺序，在圆方树中建立方点，依次向栈中的圆点连边
3. 如果一条边是树边，即没有被任何环覆盖，我们直接在圆方树中加上这条边

圆方树的性质

- 定义 + 构造了半天，我们来看看它有什么性质

圆方树的性质

- 定义 + 构造了半天，我们来看看它有什么性质

1. $\forall(x, y) \in E_T$，$\{x, y\} \cap R_T \neq \emptyset$，即两个方点不会相连

圆方树的性质

- 定义 + 构造了半天，我们来看看它有什么性质

1. $\forall (x, y) \in E_T$，$\{x, y\} \cap R_T \neq \emptyset$，即两个方点不会相连
2. 在构造过程中，无论取什么点为根，构造出的圆方树都是一样的（除了方点的编号可能不同），因此圆方树是无根树

圆方树的性质

- 定义 + 构造了半天，我们来看看它有什么性质

1. $\forall (x, y) \in E_T$，$\{x, y\} \cap R_T \neq \emptyset$，即两个方点不会相连
2. 在构造过程中，无论取什么点为根，构造出的圆方树都是一样的（除了方点的编号可能不同），因此圆方树是无根树

定义：子仙人掌

以 r 为根的仙人掌上的点 p 的子仙人掌是从仙人掌中去掉 p 到 r 的简单路径上的所有边之后，p 所在的连通块。

圆方树的性质

- 定义 + 构造了半天，我们来看看它有什么性质

1. $\forall(x,y) \in E_T$，$\{x,y\} \cap R_T \neq \emptyset$，即两个方点不会相连
2. 在构造过程中，无论取什么点为根，构造出的圆方树都是一样的（除了方点的编号可能不同），因此圆方树是无根树

定义：子仙人掌

以 r 为根的仙人掌上的点 p 的子仙人掌是从仙人掌中去掉 p 到 r 的简单路径上的所有边之后，p 所在的连通块。

3. 以 r 为根的仙人掌中点 p 的子仙人掌就是圆方树以 r 为根时点 p 的子树中的所有圆点。

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \le 1000000, m \le 2000000$

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \leq 1000000, m \leq 2000000$

- 像树那样，$f(i, 0/1)$ 表示 i 子仙人掌中 i 是否有选的最大独立集

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \leq 1000000, m \leq 2000000$

- 像树那样，$f(i, 0/1)$ 表示 i 子仙人掌中 i 是否有选的最大独立集
- 如果一条边是圆圆边（连接两个圆点），像树那样转移

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \le 1000000, m \le 2000000$

- 像树那样，$f(i, 0/1)$ 表示 i 子仙人掌中 i 是否有选的最大独立集
- 如果一条边是圆圆边（连接两个圆点），像树那样转移
- 否则（是圆方边），把这个环中所有点拿出来，跑一个环上独立集的 DP

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \le 1000000, m \le 2000000$

- 像树那样，$f(i, 0/1)$ 表示 i 子仙人掌中 i 是否有选的最大独立集
- 如果一条边是圆圆边（连接两个圆点），像树那样转移
- 否则（是圆方边），把这个环中所有点拿出来，跑一个环上独立集的 DP
- 如前面所说，其实这题不需要把圆方树建出来

在仙人掌上 DP——1

BZOJ4316

求仙人掌的最大独立集

$n \leq 1000000, m \leq 2000000$

- 像树那样，$f(i, 0/1)$ 表示 i 子仙人掌中 i 是否有选的最大独立集
- 如果一条边是圆圆边（连接两个圆点），像树那样转移
- 否则（是圆方边），把这个环中所有点拿出来，跑一个环上独立集的 DP
- 如前面所说，其实这题不需要把圆方树建出来
- 但相对于前面 DFS 树的做法，Tarjan 可以更加方便地地按顺序取出环上的节点，细节很少

在仙人掌上 DP——2

BZOJ1023

求仙人掌的直径（两点之间的最短路的最大值）

$n \leq 1000000, m \leq 2000000$

在仙人掌上 DP——2

BZOJ1023

求仙人掌的直径（两点之间的最短路的最大值）

$n \le 1000000, m \le 2000000$

- 如果一个点是圆点，可以像树那样考虑 LCA 为它的最长路径，记录这个点往下深度的最大值和次大值即可

在仙人掌上 DP——2

BZOJ1023

求仙人掌的直径（两点之间的最短路的最大值）

$n \leq 1000000, m \leq 2000000$

- 如果一个点是圆点，可以像树那样考虑 LCA 为它的最长路径，记录这个点往下深度的最大值和次大值即可
- 如果一个点是方点，我们需要以这个环为 LCA（这里是以这个方点为 LCA）的最长的最短路

在仙人掌上 DP——2

BZOJ1023

求仙人掌的直径（两点之间的最短路的最大值）

$n \leq 1000000, m \leq 2000000$

- 如果一个点是圆点，可以像树那样考虑 LCA 为它的最长路径，记录这个点往下深度的最大值和次大值即可
- 如果一个点是方点，我们需要以这个环为 LCA（这里是以这个方点为 LCA）的最长的最短路
- 经过环上的最长的最短路可以考虑经过的是环的哪一侧，用一个单调队列解决

在仙人掌上 DP——2

BZOJ1023

求仙人掌的直径（两点之间的最短路的最大值）

$n \leq 1000000, m \leq 2000000$

- 如果一个点是圆点，可以像树那样考虑 LCA 为它的最长路径，记录这个点往下深度的最大值和次大值即可
- 如果一个点是方点，我们需要以这个环为 LCA（这里是以这个方点为 LCA）的最长的最短路
- 经过环上的最长的最短路可以考虑经过的是环的哪一侧，用一个单调队列解决
- 其实也可以不把圆方树建出来，Tarjan 时候搞就行了

题外话

- Q：好像这几题都可以直接在 Tarjan 时搞，不需要把圆方树建出来啊

题外话

- Q：好像这几题都可以直接在 Tarjan 时搞，不需要把圆方树建出来啊
- A：接下来就不是这样了嘿嘿嘿

仙人掌的最短路

BZOJ2125

多次询问仙人掌两点之间最短路

$n, m, q \leq 10^6$

仙人掌的最短路

BZOJ2125

多次询问仙人掌两点之间最短路

$n, m, q \leq 10^6$

- 考虑两点之间的所有简单路径的并，一定是若干个环和若干条树边串成一串，我们能选择的就是每个环走哪一侧——由于是最短路，显然是走短的那一侧

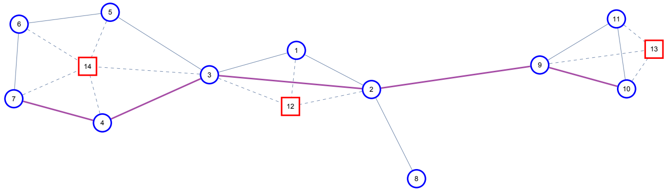
仙人掌的最短路

BZOJ2125

多次询问仙人掌两点之间最短路

$n, m, q \leq 10^6$

- 考虑两点之间的所有简单路径的并，一定是若干个环和若干条树边串成一串，我们能选择的就是每个环走哪一侧——由于是最短路，显然是走短的那一侧



仙人掌的最短路

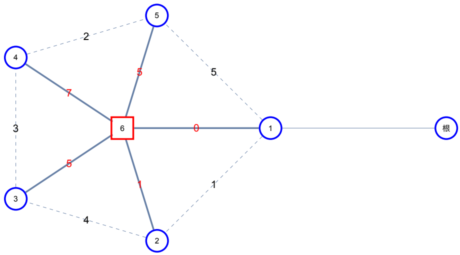
- 考虑为边设定边权，先随便取一个圆点当根，所有圆圆边的边权和原图中一致，对于每一条圆方边：

仙人掌的最短路

- 考虑为边设定边权，先随便取一个圆点当根，所有圆圆边的边权和原图中一致，对于每一条圆方边：
- 如果它是方点的父边，则定义它的边权为 0，否则定义其边权为“这个圆点到方点的父亲的最短路的长度”

仙人掌的最短路

- 考虑为边设定边权，先随便取一个圆点当根，所有圆圆边的边权和原图中一致，对于每一条圆方边：
- 如果它是方点的父边，则定义它的边权为 0，否则定义其边权为“这个圆点到方点的父亲的最短路的长度”



仙人掌的最短路

- 现在，如果两点的 LCA 是圆点，则两点的最短路就是两点的圆方树上带权距离（所有环都在已经决定了走较短一侧）

仙人掌的最短路

- 现在，如果两点的 LCA 是圆点，则两点的最短路就是两点的圆方树上带权距离（所有环都在已经决定了走较短一侧）
- 否则，我们还需要考虑 LCA 这个环走哪一侧，用树链剖分~~或倍增~~求出询问的两个点分别是在这个方点的哪两个子树中（即求出是环上的哪两个点），然后环上取较短的一侧

仙人掌的最短路

- 现在，如果两点的 LCA 是圆点，则两点的最短路就是两点的圆方树上带权距离（所有环都在已经决定了走较短一侧）
- 否则，我们还需要考虑 LCA 这个环走哪一侧，用树链剖分~~或倍增~~求出询问的两个点分别是在这个方点的哪两个子树中（即求出是环上的哪两个点），然后环上取较短的一侧
- 这样问题就完美解决了

虚仙人掌——构造

- 类比虚树，我们也可以构造虚仙人掌，可是怎么构造呢?

虚仙人掌——构造

- 类比虚树，我们也可以构造虚仙人掌，可是怎么构造呢?
- 我们要先证明一个性质

虚仙人掌——构造

- 类比虚树，我们也可以构造虚仙人掌，可是怎么构造呢?
- 我们要先证明一个性质

定理：圆方树和仙人掌等价性定理

对于任意一棵树 $T = (S_T + R_T, E_T)$，如果满足 $\forall(x, y) \in E_T$，$\{x, y\} \cap R_T \neq \emptyset$ 即两个方点不会相连，那么它是合法的圆方树，并且存在一棵仙人掌 $G = (V, E)$ 使得其圆方树为 T

虚仙人掌——构造

- 类比虚树，我们也可以构造虚仙人掌，可是怎么构造呢?
- 我们要先证明一个性质

定理：圆方树和仙人掌等价性定理

对于任意一棵树 $T=(S_T+R_T,E_T)$，如果满足 $\forall(x,y)\in E_T$，$\{x,y\}\cap R_T\neq\emptyset$ 即两个方点不会相连，那么它是合法的圆方树，并且存在一棵仙人掌 $G=(V,E)$ 使得其圆方树为 T

- 容易用构造性方法（把每个方点的出边连成环）证明

虚仙人掌——构造

- 类比虚树，我们也可以构造虚仙人掌，可是怎么构造呢?
- 我们要先证明一个性质

定理：圆方树和仙人掌等价性定理

对于任意一棵树 $T=(S_T+R_T,E_T)$，如果满足 $\forall(x,y)\in E_T$，$\{x,y\}\cap R_T\neq\emptyset$ 即两个方点不会相连，那么它是合法的圆方树，并且存在一棵仙人掌 $G=(V,E)$ 使得其圆方树为 T

- 容易用构造性方法（把每个方点的出边连成环）证明

虚仙人掌——构造

- 如果直接构造圆方树的虚树，可能会有虚边连接两个方点

虚仙人掌——构造

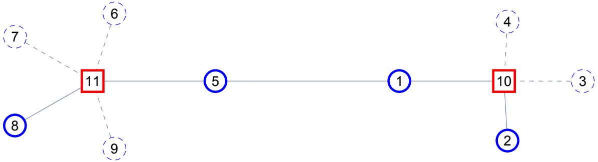
- 如果直接构造圆方树的虚树，可能会有虚边连接两个方点
- 我们可以**在所有方点连出的虚边**中间，都添加一个点表示这条出边是从环上哪一个点出去的，这样虚树的规模依然是 $O(|S|)$

虚仙人掌——构造

- 如果直接构造圆方树的虚树，可能会有虚边连接两个方点
- 我们可以**在所有方点连出的虚边**中间，都添加一个点表示这条出边是从环上哪一个点出去的，这样虚树的规模依然是 $O(|S|)$
- 下图是点集 $\{2, 8\}$ 的虚圆方树

虚仙人掌——构造

- 如果直接构造圆方树的虚树，可能会有虚边连接两个方点
- 我们可以**在所有方点连出的虚边**中间，都添加一个点表示这条出边是从环上哪一个点出去的，这样虚树的规模依然是 $O(|S|)$
- 下图是点集 $\{2, 8\}$ 的虚圆方树



- 这样构造出的虚树也是合法的圆方树，其对应的仙人掌即虚仙人掌，而大多数仙人掌问题都可以在圆方树上解决，因此虚仙人掌问题也可以在虚圆方树上解决

虚仙人掌——1

UOJ87 myy 的仙人掌

给出仙人掌，每次询问一个点集中，两个点的最短路的最大值是多少

$n, \sum |S| \leq 300000$

虚仙人掌——1

UOJ87 myy 的仙人掌

给出仙人掌，每次询问一个点集中，两个点的最短路的最大值是多少
$n, \sum |S| \le 300000$

- 建立虚圆方树，设虚边长度为树上距离

虚仙人掌——1

UOJ87 myy 的仙人掌

给出仙人掌，每次询问一个点集中，两个点的最短路的最大值是多少
$n, \sum |S| \le 300000$

- 建立虚圆方树，设虚边长度为树上距离
- 容易证明前面连接的虚边长度符合 BZOJ2125 解法所叙述的方式，问题转化为 BZOJ2125，直接做就行了

虚仙人掌——2

UOJ189 火车司机出秦川

给出仙人掌，要求资磁：

- 修改边权
- 给出一些路径，都是两点之间最短或最长简单路径的形式，求这些路径的并的边权和

这里的最长或最短路径比较的是经过边数的多少，而非边权，保证所有环都是奇环

$n, \sum |S| \leq 300000$

虚仙人掌——2

UOJ189 火车司机出秦川

给出仙人掌，要求资磁：

- 修改边权
- 给出一些路径，都是两点之间最短或最长简单路径的形式，求这些路径的并的边权和

这里的最长或最短路径比较的是经过边数的多少，而非边权，保证所有环都是奇环

$n, \sum |S| \leq 300000$

- 如果是树上的链并，一种方式是树链剖分，但复杂度是 $O(\log^2 n)$ 的；另一种是虚树，转为求虚树上被至少覆盖一次的边的边权和，这可以用前缀和轻松解决

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可
- 对于所有的圆圆虚边，对应的是仙人掌上两点之间的简单路径的并，其被覆盖的情况有 4 种：

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可
- 对于所有的圆圆虚边，对应的是仙人掌上两点之间的简单路径的并，其被覆盖的情况有 4 种：
  1. 全都没有被覆盖
  2. 只有最短路径被覆盖
  3. 只有最长路径被覆盖
  4. 全都被覆盖

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可
- 对于所有的圆圆虚边，对应的是仙人掌上两点之间的简单路径的并，其被覆盖的情况有 4 种：
  1. 全都没有被覆盖
  2. 只有最短路径被覆盖
  3. 只有最长路径被覆盖
  4. 全都被覆盖
- 我们可以对于每一个点，维护它的三种深度：

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可
- 对于所有的圆圆虚边，对应的是仙人掌上两点之间的简单路径的并，其被覆盖的情况有 4 种：
  1. 全都没有被覆盖
  2. 只有最短路径被覆盖
  3. 只有最长路径被覆盖
  4. 全都被覆盖
- 我们可以对于每一个点，维护它的三种深度：
  1. *depTree*：从根到它经过的树边的边权和
  2. *depMin*：从根到它经过的每个环较短一侧的边权和
  3. *depMax*：从根到它经过的每个环较长一侧的边权和

虚仙人掌——2

- 现在是仙人掌，我们仍然考虑建立虚圆方树；对每个环，被覆盖的是若干个区间，利用前缀和可以求出一这些区间，排序求并即可
- 对于所有的圆圆虚边，对应的是仙人掌上两点之间的简单路径的并，其被覆盖的情况有 4 种：
  1. 全都没有被覆盖
  2. 只有最短路径被覆盖
  3. 只有最长路径被覆盖
  4. 全都被覆盖
- 我们可以对于每一个点，维护它的三种深度：
  1. *depTree*：从根到它经过的树边的边权和
  2. *depMin*：从根到它经过的每个环较短一侧的边权和
  3. *depMax*：从根到它经过的每个环较长一侧的边权和
- 那么 4 种覆盖情况都能表示为某几种深度的和

虚仙人掌——2

- 现在如何资磁修改

虚仙人掌——2

- 现在如何资磁修改
- 首先环上的并要求环的区间和，用树状数组可以维护

虚仙人掌——2

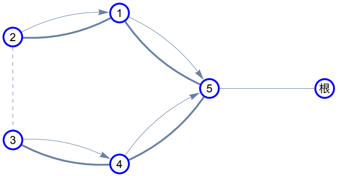
- 现在如何资磁修改
- 首先环上的并要求环的区间和，用树状数组可以维护
- 考虑维护三种深度，如果修改的是树边，直接子树的 $depTree$ 加上差值

虚仙人掌——2

- 现在如何资磁修改
- 首先环上的并要求环的区间和，用树状数组可以维护
- 考虑维护三种深度，如果修改的是树边，直接子树的 *depTree* 加上差值
- 对于每一个奇环，我们将它从方点的父亲剖开后，取其中间这条边作为分界，那么它左边的点的最短路是往左走，右边的点的最短路是往右走

虚仙人掌——2

- 现在如何资磁修改
- 首先环上的并要求环的区间和，用树状数组可以维护
- 考虑维护三种深度，如果修改的是树边，直接子树的 *depTree* 加上差值
- 对于每一个奇环，我们将它从方点的父亲剖开后，取其中间这条边作为分界，那么它左边的点的最短路是往左走，右边的点的最短路是往右走



虚仙人掌——2

- 那么对于每一次修改，一定是分界点某一侧一个区间的子树中的 $depMin$ 加一个数，另一侧的一个区间的子树中的 $depMax$ 加一个数，树状数组维护即可

虚仙人掌——2

- 那么对于每一次修改，一定是分界点某一侧一个区间的子树中的 *depMin* 加一个数，另一侧的一个区间的子树中的 *depMax* 加一个数，树状数组维护即可
- 这样问题就完美解决了

仙人掌分治

UOJ23 跳蚤国王下江南

给出一张仙人掌，要求对 $i \in [1, n)$ 输出从 1 出发的长度为 i 的简单路径有多少条

$n \leq 100000$

仙人掌分治

- 仙人掌上，经过一个环中每两个点都有两种走法

仙人掌分治

- 仙人掌上，经过一个环中每两个点都有两种走法
- 设 $g_p(z)$ 表示点 p 往子树中的所有路径的生成函数，则对于一个长度为 $r+1$ 的环 $R_1, R_2, ..., R_r$（这里是按顺序列举出所有非环根的节点），其贡献为

$$\sum_{i=1}^{r} g_{R_i}(z)(z^i + z^{r-i+1})$$

仙人掌分治

- 仙人掌上，经过一个环中每两个点都有两种走法
- 设 $g_p(z)$ 表示点 p 往子树中的所有路径的生成函数，则对于一个长度为 $r+1$ 的环 $R_1, R_2, \dots, R_r$（这里是按顺序列举出所有非环根的节点），其贡献为

$$\sum_{i=1}^{r} g_{R_i}(z)(z^i + z^{r-i+1})$$

- 我们可以进行分治——对圆方树进行点分治。

仙人掌分治

- 记 $G_p(z)$ 表示以 p 为重心的连通子树中，从该连通子树的根往下的不同路径的方案数的生成函数，设当前连通块的重心为 g，g 到根的分支的重心为 p（如果 g 就是根，则不存在 p，不妨设 $G_p(z)=0$），g 到根这一段的路径的生成函数为 $F_g(z)$，g 除掉 p 的其他分支的重心集合为 S_g

仙人掌分治

- 记 $G_p(z)$ 表示以 p 为重心的连通子树中，从该连通子树的根往下的不同路径的方案数的生成函数，设当前连通块的重心为 g，g 到根的分支的重心为 p（如果 g 就是根，则不存在 p，不妨设 $G_p(z)=0$），g 到根这一段的路径的生成函数为 $F_g(z)$，g 除掉 p 的其他分支的重心集合为 S_g
- 如果重心 g 是圆点，则

$$G_g(z) = G_p(z) + F_g(z) \sum_{q \in S_g} G_q(z)$$

仙人掌分治

- 记 $G_p(z)$ 表示以 p 为重心的连通子树中，从该连通子树的根往下的不同路径的方案数的生成函数，设当前连通块的重心为 g，g 到根的分支的重心为 p（如果 g 就是根，则不存在 p，不妨设 $G_p(z)=0$），g 到根这一段的路径的生成函数为 $F_g(z)$，g 除掉 p 的其他分支的重心集合为 S_g
- 如果重心 g 是圆点，则

$$G_g(z)=G_p(z)+F_g(z)\sum_{q\in S_g}G_q(z)$$

- 否则设点 q 是 g 环中第 $id_g(q)$ 个点，则

$$G_g(z)=G_p(z)+F_g(z)\sum_{q\in S_g}G_q(z)(z^{id_g(q)}+z^{r-id_g(q)+1})$$

仙人掌分治

- 首先我们需要求出 $F_g(z)$，直接分治 FFT 是 $O(n\log^3 n)$ 的，如果合并在点分治中做可以优化到 $O(n\log^2 n)$，这里限于篇幅就不讲了

仙人掌分治

- 首先我们需要求出 $F_g(z)$，直接分治 FFT 是 $O(n\log^3 n)$ 的，如果合并在点分治中做可以优化到 $O(n\log^2 n)$，这里限于篇幅就不讲了
- 在计算生成函数时，圆点可以直接 FFT 模拟多项式乘法，但是方点这么做复杂度是错误的，因为如果有一个 $O(n)$ 的环，

$$\sum_{q\in S_g} G_q(z)(z^{id_g(q)} + z^{r-id_g(q)+1})$$

中每一项的次数都是 $O(n)$ 的

仙人掌分治

- 处理方式很简单，注意到

$$\sum_{q\in S_g} G_q(z)(z^{id_g(q)} + z^{r-id_g(q)+1})$$

$$= \sum_{q\in S_g} G_q(z)z^{id_g(q)} + \sum_{q\in S_g} G_q(z)z^{r-id_g(q)+1}$$

仙人掌分治

- 处理方式很简单，注意到

$$\sum_{q\in S_g} G_q(z)(z^{id_g(q)}+z^{r-id_g(q)+1})$$

$$=\sum_{q\in S_g} G_q(z)z^{id_g(q)}+\sum_{q\in S_g} G_q(z)z^{r-id_g(q)+1}$$

- 每一项都是多项式乘 z^k 的形式

仙人掌分治

- 处理方式很简单，注意到

$$\sum_{q\in S_g} G_q(z)(z^{id_g(q)} + z^{r-id_g(q)+1})$$

$$= \sum_{q\in S_g} G_q(z)z^{id_g(q)} + \sum_{q\in S_g} G_q(z)z^{r-id_g(q)+1}$$

- 每一项都是多项式乘 z^k 的形式
- 因此我们只需要实现一个移位后多项式加法即可

仙人掌分治

- 处理方式很简单，注意到

$$\sum_{q\in S_g} G_q(z)(z^{id_g(q)} + z^{r-id_g(q)+1})$$

$$= \sum_{q\in S_g} G_q(z)z^{id_g(q)} + \sum_{q\in S_g} G_q(z)z^{r-id_g(q)+1}$$

- 每一项都是多项式乘 z^k 的形式
- 因此我们只需要实现一个移位后多项式加法即可
- 这样问题就在 $O(n\log^2 n)$ 的时间内完美解决了，实际运行效率非常优秀

仙人掌剖分

UOJ158 静态仙人掌（数据范围扩大版）

给出一棵根为 1 的仙人掌，每个点是黑色或白色，保证所有环都是奇环，要求资磁三种操作：

1. 把一个点到根的最短路径上的所有点颜色取反
2. 把一个点到根的最长简单路径上的所有点颜色取反
3. 询问一个点的子仙人掌里面有多少个黑点

$n, q \leq 200000$

仙人掌剖分

- 如果是树上的问题，那我们很容易用树链剖分套线段树解决

仙人掌剖分

- 如果是树上的问题，那我们很容易用树链剖分套线段树解决
- 现在是仙人掌上的问题，由圆方树的性质，子仙人掌就是圆方树的子树，因此我们就只需要考虑如何进行修改

仙人掌剖分

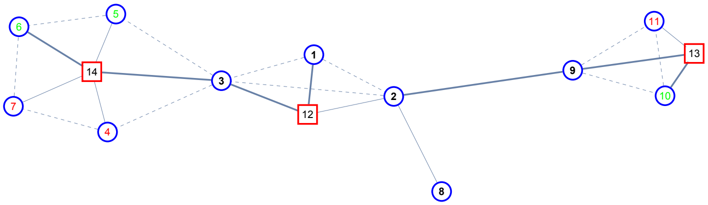
- 如果是树上的问题，那我们很容易用树链剖分套线段树解决
- 现在是仙人掌上的问题，由圆方树的性质，子仙人掌就是圆方树的子树，因此我们就只需要考虑如何进行修改
- 借用树剖的思想，我们需要支持的是：对一条重链执行快速修改。具体地来说，是要求资磁对一条重链的一个前缀中的最长或最短路径进行快速修改。

仙人掌剖分

- 我们同样考虑将圆点进行分类:
  1. 必须经过的点（即割点，加粗）
  2. 在这条重链上，只出现在最短路上的点（绿色）
  3. 在这条重链上，只出现在最长简单路径上的点（红色）

仙人掌剖分

- 我们同样考虑将圆点进行分类:
  1. 必须经过的点（即割点，加粗）
  2. 在这条重链上，只出现在最短路上的点（绿色）
  3. 在这条重链上，只出现在最长简单路径上的点（红色）



仙人掌剖分

- 这样每次修改只需要修改这条重链上的点（如果是方点，其连出圆点也要考虑）中的某若干类点即可

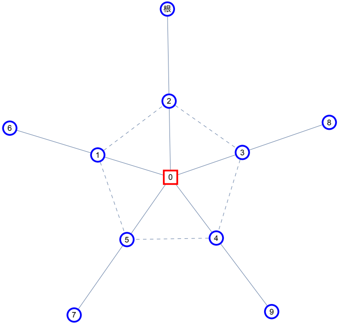
仙人掌剖分

- 这样每次修改只需要修改这条重链上的点（如果是方点，其连出圆点也要考虑）中的某若干类点即可
- 传统的树链剖分序并不支持考虑方点连出的圆点，因此我们重新定义方点连出的 DFS 序

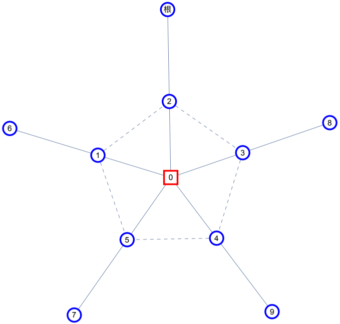
仙人掌剖分

- 这样每次修改只需要修改这条重链上的点（如果是方点，其连出圆点也要考虑）中的某若干类点即可
- 传统的树链剖分序并不支持考虑方点连出的圆点，因此我们重新定义方点连出的 DFS 序
- 对于每个方点，现在 DFS 序中加入其连出的所有圆点，然后再递归访问每个圆点的子树

仙人掌剖分

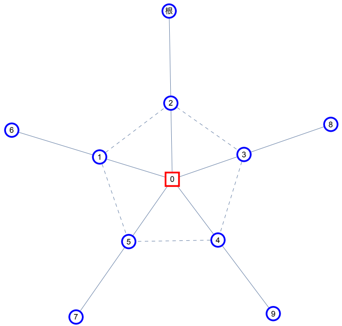


仙人掌剖分



- 上图的 DFS 序为：根,2,0,**1,5,4,3**,6,7,9,8

仙人掌剖分



- 上图的 DFS 序为：根,2,0,**1,5,4,3**,6,7,9,8
- 这样我们就能在链修改时访问到所有点了，把每一类点分别用线段树维护即可，问题完美解决

diff 缩环树 圆方树

- 既然讲了这几道题，我们便需要开始做个比较了

diff 缩环树 圆方树

- 既然讲了这几道题，我们便需要开始做个比较了
- 去年小火车在国冬上使用“缩环树”这一结构解决了《静态仙人掌》一题，并使用此结构出了《火车司机出秦川》

diff 缩环树 圆方树

- 既然讲了这几道题，我们便需要开始做个比较了
- 去年小火车在国冬上使用“缩环树”这一结构解决了《静态仙人掌》一题，并使用此结构出了《火车司机出秦川》
- 那么缩环树和圆方树之间有什么联系和区别呢?

diff 缩环树 圆方树

- 共同点：
  1. 都是把仙人掌转化为树来做

diff 缩环树 圆方树

- 共同点:
  1. 都是把仙人掌转化为树来做
  2. 同样可以剖分和建立虚树

diff 缩环树 圆方树

- 共同点：
  1. 都是把仙人掌转化为树来做
  2. 同样可以剖分和建立虚树
- 不同点：
  1. 缩环树是有根树，相当于圆方树取一个根之后把父亲为方点的圆点缩掉，圆方树是无根树

diff 缩环树 圆方树

- 共同点:
  1. 都是把仙人掌转化为树来做
  2. 同样可以剖分和建立虚树
- 不同点:
  1. 缩环树是有根树，相当于圆方树取一个根之后把父亲为方点的圆点缩掉，圆方树是无根树
  2. 每棵圆方树都存在一个等价的仙人掌，而可能会有多个仙人掌的缩环树相同

diff 缩环树 圆方树

- 共同点：
  1. 都是把仙人掌转化为树来做
  2. 同样可以剖分和建立虚树
- 不同点：
  1. 缩环树是有根树，相当于圆方树取一个根之后把父亲为方点的圆点缩掉，圆方树是无根树
  2. 每棵圆方树都存在一个等价的仙人掌，而可能会有多个仙人掌的缩环树相同
  3. 查询一个点是环中哪个子仙人掌的点时，缩环树需在 DFS 序中二分，圆方树可用树链剖分来跳跃（也可以二分）

diff 缩环树 圆方树

- 共同点：
  1. 都是把仙人掌转化为树来做
  2. 同样可以剖分和建立虚树
- 不同点：
  1. 缩环树是有根树，相当于圆方树取一个根之后把父亲为方点的圆点缩掉，圆方树是无根树
  2. 每棵圆方树都存在一个等价的仙人掌，而可能会有多个仙人掌的缩环树相同
  3. 查询一个点是环中哪个子仙人掌的点时，缩环树需在 DFS 序中二分，圆方树可用树链剖分来跳跃（也可以二分）
- 本猫认为，正因为圆方树这一结构没有缩点，它才能获得更好的性质——点对应、无根性和仙人掌等价性，能更方便的胜任虚仙人掌和分治的题目（神犇轻 D）

点双连通分量题目的新思考

- 如果我们在点双分量的题目中，对每个点双连通分量建立“方点”，然后向双连通分量中每个点连边，这样也可以构造出一种“圆方树”，我们称这种圆方树为“广义圆方树”

点双连通分量题目的新思考

- 如果我们在点双分量的题目中，对每个点双连通分量建立“方点”，然后向双连通分量中每个点连边，这样也可以构造出一种“圆方树”，我们称这种圆方树为“广义圆方树”
- 根据圆方树和仙人掌等价性定理，我们可以为一张一般图建立一棵“等效仙人掌”(环上的点的顺序是任意的)——一个环可以代表一个任意点双连通分量，这可能有助于我们理解题意，更好地思考算法

广义圆方树——1

商人

给出一张图，点有点权。每次询问两点之间的简单路径中，权值的最小值最小是多少。

$n, m, q \leq 1000000$

广义圆方树——1

商人

给出一张图，点有点权。每次询问两点之间的简单路径中，权值的最小值最小是多少。

$n, m, q \leq 1000000$

- 直接考虑点双连通分量并不方便（因为路径太多了），我们考虑建立圆方树，转化为等效仙人掌

广义圆方树——1

商人

给出一张图，点有点权。每次询问两点之间的简单路径中，权值的最小值最小是多少。
$n, m, q \leq 1000000$

- 直接考虑点双连通分量并不方便（因为路径太多了），我们考虑建立圆方树，转化为等效仙人掌
- 这样问题就很容易理解了，转化为两点所有简单路径的并中，权值的最小值

广义圆方树——1

商人

给出一张图，点有点权。每次询问两点之间的简单路径中，权值的最小值最小是多少。

$n, m, q \leq 1000000$

- 直接考虑点双连通分量并不方便（因为路径太多了），我们考虑建立圆方树，转化为等效仙人掌
- 这样问题就很容易理解了，转化为两点所有简单路径的并中，权值的最小值
- 只需要将每个方点的权值设为其连出所有圆点的最小值，这样问题就直接转化为了链上最小值

广义圆方树——1

商人

给出一张图，点有点权。每次询问两点之间的简单路径中，权值的最小值最小是多少。

$n, m, q \leq 1000000$

- 直接考虑点双连通分量并不方便（因为路径太多了），我们考虑建立圆方树，转化为等效仙人掌
- 这样问题就很容易理解了，转化为两点所有简单路径的并中，权值的最小值
- 只需要将每个方点的权值设为其连出所有圆点的最小值，这样问题就直接转化为了链上最小值
- 并查集就行了（NOIP 难度）

广义圆方树——2

Codechef SADPAIRS

给出一张图，每个点有颜色。对于每个点求出，如果删掉这个点，不连通的同色点对有几个。

$n, m, q \leq 10^7$

广义圆方树——2

Codechef SADPAIRS

给出一张图，每个点有颜色。对于每个点求出，如果删掉这个点，不连通的同色点对有几个。
$n, m, q \leq 10^7$

- 分别考虑每一种颜色计算这种颜色不连通带来的贡献，对每一种颜色建立圆方树的虚树

广义圆方树——2

Codechef SADPAIRS

给出一张图，每个点有颜色。对于每个点求出，如果删掉这个点，不连通的同色点对有几个。
$n, m, q \leq 10^7$

- 分别考虑每一种颜色计算这种颜色不连通带来的贡献，对每一种颜色建立圆方树的虚树
- 对于虚树上每一条边，对应的是圆方树上的一条链，其贡献均为子树内点数乘子树外点数，差分维护；对于虚树上的每一个点，贡献为路径经过它的这种颜色的点对数，直接计算

广义圆方树——2

Codechef SADPAIRS

给出一张图，每个点有颜色。对于每个点求出，如果删掉这个点，不连通的同色点对有几个。
$n, m, q \leq 10^7$

- 分别考虑每一种颜色计算这种颜色不连通带来的贡献，对每一种颜色建立圆方树的虚树
- 对于虚树上每一条边，对应的是圆方树上的一条链，其贡献均为子树内点数乘子树外点数，差分维护；对于虚树上的每一个点，贡献为路径经过它的这种颜色的点对数，直接计算
- 用 RMQ 线性求 LCA，虚树排序用基数排序，复杂度为线性

这东西有什么用

- 更好更通用~~更无脑~~地解决一些仙人掌题，降低仙人掌题难度

这东西有什么用

- 更好更通用~~更无脑~~地解决一些仙人掌题，降低仙人掌题难度
- 联系仙人掌和点双连通分量，使我们想题时能更加具体

这东西有什么用

- 更好更通用~~更无脑~~地解决一些仙人掌题，降低仙人掌题难度
- 联系仙人掌和点双连通分量，使我们想题时能更加具体
- 将点双连通分量~~和仙人掌~~的题发扬光大

这东西有什么用

- 更好更通用~~更无脑~~地解决一些仙人掌题，降低仙人掌题难度
- 联系仙人掌和点双连通分量，使我们想题时能更加具体
- 将点双连通分量~~和仙人掌~~的题发扬光大，出到 NOIP 中去

展望

- 大家能想出更多、更好的仙人掌问题的 idea

展望

- 大家能想出更多、更好的仙人掌问题的 idea
- 圆方树能够得到发展，能够适用于动态仙人掌问题

感谢

- 感谢父母的养育之恩、陈颖老师和余林韵、陈许旻、张瑞喆等教练和董克凡学长的栽培

感谢

- 感谢父母的养育之恩、陈颖老师和余林韵、陈许旻、张瑞喆等教练和董克凡学长的栽培
- 感谢福州一中的刘一凡（LGG）同学和我一起想出了“圆方树”这一名称，并补充了一些细节，在我写《静态仙人掌》和《火车司机出秦川》时帮我检查代码

感谢

- 感谢父母的养育之恩、陈颖老师和余林韵、陈许旻、张瑞喆等教练和董克凡学长的栽培
- 感谢福州一中的刘一凡（LGG）同学和我一起想出了“圆方树”这一名称，并补充了一些细节，在我写《静态仙人掌》和《火车司机出秦川》时帮我检查代码

完结撒花 GL&HF

让常数优化成为一种习惯！

WC2017 ++RP