树相关

By zhan8855

目录

- 1 树的基础算法
- 2 树上信息的维护
- 3 树的计数
- 4 无向图的生成树

1 树的基础算法

目录

- 1.1 基本性质
- 1.2 重心
- *1.2 点分治
- 1.3 最长链
- 1.4 最近公共祖先
- *1.4 虚树

1.1 基本性质

1.1.1 定义

- 树是一张由V个点、 V-1条边组成的连通图。

1.1.2 例题选讲

- [Codeforces453C] Little Pony and Summer Sun Celebration
- 题意：
- 给出一张N个点M条边的无向图，有些点要求经过奇数次，有些点要求经过偶数次，要求寻找一条满足要求的路径，且该路径长度不超过点数的四倍。
- N, M≤100000

1.1.2 例题选讲

- [Codeforces453C] Little Pony and Summer Sun Celebration
- 题意：
- 给出一张N个点M条边的无向图，有些点要求经过奇数次，有些点要求经过偶数次，要求寻找一条满足要求的路径，且该路径长度不超过点数的四倍。
- N, M≤100000
- 标准算法：
- 求出图的任意生成树，任意定根，搜索。很明显，叶子必定只访问一次，非叶结点必定只访问两次。
- 但是，如果非叶结点要求访问奇数次，上面的做法就会出现问题。这时只要让它的父亲多遍历这个点一次即可。
- 但是，如果这个点没有父亲呢？这样的点有且仅可能有一个，即树根。这样只要对一个奇数点回溯即可。

1.2 重心

1.2.1 定义

- 树的重心是到树上所有点距离之和最小的点。

1.2.1 定义

- 树的重心是到树上所有点距离之和最小的点。
- 重心的充要条件：最大的子树大小不超过全树的一半。

1.2.1 定义

- 树的重心是到树上所有点距离之和最小的点。
- 重心的充要条件：最大的子树大小不超过全树的一半。
- 通常可以一遍DFS求解。

1.2.2 例题选讲

- [Codeforces709E] Centroids
- 题意：
- 给出一棵N个点的树，对于树的所有节点，询问如果删掉一条边再加上一条边，能否将其改为重心。
- N≤400000

1.2.2 例题选讲

- [Codeforces709E] Centroids
- 题意：
- 给出一棵N个点的树，对于树的所有节点，询问如果删掉一条边再加上一条边，能否将其改为重心。
- N≤400000

- 标准算法：
- 考虑重心的充要条件：不存在任何一棵子树的点数超过总点数的一半。因此，对于每个非重心节点，减掉的那棵子树，必定是在包含重心子树中；进一步分析，必定是以重心直接儿子为根的一棵子树；更进一步分析，必定是重心的重儿子。
- 因此，对树遍历就可以解决这个问题了。

*1.2 点分治

*1.2.1 方法概述

- 点分治的一般思路是寻找树的重心、删去重心把树分为若干部分、再寻找各部分重心……的递归过程，并在各个重心处统计答案。

*1.2.1 方法概述

- 点分治的一般思路是寻找树的重心、删去重心把树分为若干部分、再寻找各部分重心……的递归过程，并在各个重心处统计答案。

- 通常用于全树链计数。

*1.2.2 例题选讲

- [BZOJ3697] 采药人的路径
- 题意：
- 给出一棵N个点的树，边权为0或1。求能被划分成两段、每段0的个数与1的个数相等的简单路径数。
- N≤100000

*1.2.2 例题选讲

- [BZOJ3697] 采药人的路径
- 题意：
- 给出一棵N个点的树，边权为0或1。求能被划分成两段、每段0的个数与1的个数相等的简单路径数。
- N≤100000

- 标准算法：
- 维护f0[i]和f1[i]，表示1的个数比0的个数多i的情况下，路径上的某一段（不）可以构成一段子路径的路径总数。DFS传递i时可以用桶判断一条链上是否可以截下子路径。注意特判分界点在根节点的情况。

1.3 最长链

1.3.1 定义

- 最长链又称直径，是连接树上最远点对的路径。

1.3.1 定义

- 最长链又称直径，是连接树上最远点对的路径。
- 通常可以两遍BFS或者一遍DFS求解。

1.3.1 定义

- 最长链又称直径，是连接树上最远点对的路径。
- 通常可以两遍BFS或者一遍DFS求解。
- 注意如果树有负权边，前一种方法会出错。

1.3.2 例题选讲

- [51nod1766] 树上最远点对
- 题意：
- 给出一棵N个点的树，Q次询问一点编号在区间[l1,r1]内，另一点编号在区间[l2,r2]内的所有点对距离最大值。
- N, Q≤100000

1.3.2 例题选讲

- [51nod1766] 树上最远点对
- 题意：
- 给出一棵N个点的树，Q次询问一点编号在区间[l1,r1]内，另一点编号在区间[l2,r2]内的所有点对距离最大值。
- N, Q≤100000

- 标准算法：
- 由直径的证明可以得到，设区间[l1,r1]内的直径为ab，区间[l2,r2]内的直径为cd，区间[l1,r1]和区间[l2,r2]内任取两点作为两端点形成的直径，必定是ab、ac、ad、bc、bd、cd中的一条。因此，可以用线段树来维护。

1.4 最近公共祖先

1.4.1 定义

- 最近公共祖先简称LCA。在一棵有根树中，连接两点的路径上深度最浅的点就是这两点的LCA。

1.4.1 定义

- 最近公共祖先简称LCA。在一棵有根树中，连接两点的路径上深度最浅的点就是这两点的LCA。

- 快速求LCA的方法：
- 倍增算法
- Tarjan算法
- DFS+ST算法
- 树链剖分算法

1.4.1 定义

- 最近公共祖先简称LCA。在一棵有根树中，连接两点的路径上深度最浅的点就是这两点的LCA。

- 快速求LCA的方法：
- 倍增算法（在线，可以维护较多信息，支持加叶子）
- Tarjan算法（离线，局限性较大）
- DFS+ST算法（在线，可以用±1RMQ优化到O(1)）
- 树链剖分算法（在线，速度较快）

1.4.2 例题选讲

- [BZOJ1776] Cow Politics
- 题意：
- 给出一棵N个点的树，树上每个节点都有一种颜色。对于每种颜色，求该颜色距离最远的两个点之间的距离。
- N≤200000

1.4.2 例题选讲

- [BZOJ1776] Cow Politics
- 题意：
- 给出一棵N个点的树，树上每个节点都有一种颜色。对于每种颜色，求该颜色距离最远的两个点之间的距离。
- N≤200000

- 标准算法一：
- 对于每种颜色，如果节点数量不超过根号节点数则计算两两的LCA，否则遍历全树求叙述直径。总时间复杂度O(N^1.5)。

1.4.2 例题选讲

- [BZOJ1776] Cow Politics
- 题意：
- 给出一棵N个点的树，树上每个节点都有一种颜色。对于每种颜色，求该颜色距离最远的两个点之间的距离。
- N≤200000

- 标准算法二：
- 任意定根，考虑距离最远的两个点，必然有一个点是该颜色在树上位置最深的点。因此，找到每种颜色位置最深的点后，暴力求LCA即可。总时间复杂度O(N log N)。

*1.4 虚树

*1.4.1 方法概述

- 当问题只涉及树上若干个点的关系时，可以考虑用这些点建立虚树。

*1.4.1 方法概述

- 当问题只涉及树上若干个点的关系时，可以考虑用这些点建立虚树。

- 可以把所有点按照DFS序排序后对所有相邻点对两两求LCA，再根据DFS序的包含关系建立虚树。

*1.4.1 方法概述

- 当问题只涉及树上若干个点的关系时，可以考虑用这些点建立虚树。
- 可以把所有点按照DFS序排序后对所有相邻点对两两求LCA，再根据DFS序的包含关系建立虚树。
- 根据K个点建立的虚树点数是O(K)级别的，建立虚树的复杂度可以做到O(K log K)。虚树规模比原树大大缩小，在某些问题中可以大幅提升效率。

*1.4.2 例题选讲

- [Codeforces639F] Bear and Chemistry
- 题意：
- 给出一张图，每次询问给出若干个点和若干条边，要求判断加入给出边后，给出点是否在同一个点双连通分量中。图的点数、边数与询问总点数、总边数均不超过300000。

*1.4.2 例题选讲

- [Codeforces639F] Bear and Chemistry
- 题意：
- 给出一张图，每次询问给出若干个点和若干条边，要求判断加入给出边后，给出点是否在同一个点双连通分量中。图的点数、边数与询问总点数、总边数均不超过300000。

- 标准算法：
- 首先用Tarjan把图转化成树。每次询问求出虚树之后再跑Tarjan。

1 更多练习（基础）

- [BZOJ1787] 紧急集合
- [BZOJ1912] 巡逻
- [BZOJ2152] 聪聪可可
- [BZOJ3252] 攻略
- [BZOJ3697] 采药人的路径
- [Codeforces685B] Kay and Snowflake
- [Codeforces700B] Connecting Universities
- [Codeforces708C] Centroids
- [Codeforces715C] Digit Tree
- [Codeforces776F] Sherlock's bet to Moriarty
- [Codeforces519E] A and B and Lecture Rooms
- [Codeforces593D] Happy Tree Party

1 更多练习（中等）

- [51nod1766] 树上最远点对
- [51Nod1681] 公共祖先
- [Codeforces741D] Arpa’s letter-marked tree and Mehrdad’s Dokhtar-kosh paths
- [Codeforces633G] Yash And Trees
- [Codeforces639F] Bear and Chemistry
- [Codeforces562A] Logistical Questions

2 树上信息的维护

目录

- 2.1 倍增法
- 2.2 差分法
- 2.3 DFS序
- 2.4 重链剖分
- 2.5 动态树
- 2.6 点分树
- 2.7 树上莫队

2.1 倍增法

2.1.1 方法概述

- 倍增法的一般思路是在每个点上开一个长度为log N的数组，第i个元素维护该点出发到该点的2^i倍祖先的链上信息。查询时把链<u, v>拆成<u, LCA>和<LCA, v>两部分，分别从u和v出发向上跳跃。

2.1.1 方法概述

- 倍增法的一般思路是在每个点上开一个长度为log N的数组，第i个元素维护该点出发到该点的2^i倍祖先的链上信息。查询时把链<u, v>拆成<u, LCA>和<LCA, v>两部分，分别从u和v出发向上跳跃。

- 倍增法使用的一般条件：
- 1、修改操作很少或没有
- 2、信息可合并

2.1.2 例题选讲

- [Codeforces 609E] Minimum spanning tree for each edge
- 题意：
- 给出一张N个点M条边无向图，依次强制每一条边在生成树上，要求生成树的边权和最小，输出边权和。
- N, M≤200000

2.1.2 例题选讲

- [Codeforces 609E] Minimum spanning tree for each edge
- 题意：
- 给出一张N个点M条边无向图，依次强制每一条边在生成树上，要求生成树的边权和最小，输出边权和。
- N, M≤200000

- 标准算法：
- 如果是最小生成树上的边，直接输出最小生成树的边权和即可。
- 如果不是最小生成树上的边，把它加入最小生成树会形成环，删去环上次大的边即可。

2.2 差分法

2.2.1 方法概述

- 树上差分的一般步骤是把链<u, v>拆成<u, LCA>和<LCA, v>两部分，每部分分别在两端打上标记，最后把全树DFS一遍，回溯时统计答案。

2.2.1 方法概述

- 树上差分的一般步骤是把链<u, v>拆成<u, LCA>和<LCA, v>两部分，每部分分别在两端打上标记，最后把全树DFS一遍，回溯时统计答案。

- 差分法使用的一般条件：
- 1、支持离线
- 2、信息可合并
- 3、信息可拆分

2.2.2 例题选讲

- [BZOJ4326] 运输计划
- 题意：
- 给出一棵N个点的树和M个询问，树边有边权，每次询问答案为树上两点之间边的权值和。现在可以把其中一条边权值改为零，问如何修改才能使最大的询问答案尽量小。
- N, M≤300000

2.2.2 例题选讲

- [BZOJ4326] 运输计划
- 题意：
- 给出一棵N个点的树和M个询问，树边有边权，每次询问答案为树上两点之间边的权值和。现在可以把其中一条边权值改为零，问如何修改才能使最大的询问答案尽量小。
- N, M≤300000

- 标准算法：
- 二分边权，对边权大于二分结果的询问求一个交，判断交中是否有一条边，使得答案最大的询问减去这条边的边权结果小于二分的值。求询问答案可以用倍增法来完成，求交可以用树上差分法来实现。

2.2.2 例题选讲

- [BZOJ3307] 雨天的尾巴
- 题意：
- 给出一棵N个点的树，M次操作在链上加上某一种类别的物品，完成所有操作后，要求询问每个点上最多物品的类型。
- N, M≤100000

2.2.2 例题选讲

- [BZOJ3307] 雨天的尾巴
- 题意：
- 给出一棵N个点的树，M次操作在链上加上某一种类别的物品，完成所有操作后，要求询问每个点上最多物品的类型。
- N, M≤100000

- 标准算法：
- 树上差分，统计答案时用合并权值线段树来维护。

2.3 DFS序

2.3.1 方法概述

- DFS序最常见的用途是维护子树信息，有时也可以维护链信息。

2.3.1 方法概述

- DFS序最常见的用途是维护子树信息，有时也可以维护链信息。
- 常用的维护子树信息的DFS序有入栈序和出栈序。

2.3.1 方法概述

- DFS序最常见的用途是维护子树信息，有时也可以维护链信息。
- 常用的维护子树信息的DFS序有入栈序和出栈序。
- 维护树链上的信息的DFS序一般有两种。一种是入栈出栈序，另一种是括号序列。两者的共同点是都利用差分抵消了其他子树的影响，从而使DFS区间能够维护链的信息。

2.3.1 方法概述

- DFS序最常见的用途是维护子树信息，有时也可以维护链信息。
- 常用的维护子树信息的DFS序有入栈序和出栈序。
- 维护树链上的信息的DFS序一般有两种。一种是入栈出栈序，另一种是括号序列。两者的共同点是都利用差分抵消了其他子树的影响，从而使DFS区间能够维护链的信息。
- 入栈出栈序一般要求信息可合并，可以支持子树修改，不能支持链修改。
- 括号序列实现比较简单，可以维护的信息也比较少，但适用于有些特殊场合。

2.3.1 方法概述

- DFS序最常见的用途是维护子树信息，有时也可以维护链信息。
- 常用的维护子树信息的DFS序有入栈序和出栈序。
- 维护树链上的信息的DFS序一般有两种。一种是入栈出栈序，另一种是括号序列。两者的共同点是都利用差分抵消了其他子树的影响，从而使DFS区间能够维护链的信息。
- 入栈出栈序一般要求信息可合并，可以支持子树修改，不能支持链修改。
- 括号序列实现比较简单，可以维护的信息也比较少，但适用于有些特殊场合。
- 另外，欧拉序是一种特殊的DFS序，典型的应用场合是DFS+ST求LCA。

2.3.2 例题选讲

- [51Nod1681] 公共祖先
- 题意：
- 给出N个点的树，对于所有点对求它们在两棵树中公共的公共祖先数量之和。
- $N \le 100000$

2.3.2 例题选讲

- [51Nod1681] 公共祖先
- 题意：
- 给出N个点的树，对于所有点对求它们在两棵树中公共的公共祖先数量之和。
- N≤100000
- 标准算法：
- 首先，题意可以转化为对于每个点，求有多少个点，在两棵树中都是它的祖先。统计出这些之后，只需要计算组合数就可以得到最终答案。
- 要想在两棵树中同步统计答案，就需要想办法把这两部分的答案综合起来。如果要统计一棵树上某一点为多少点的祖先，既可以用搜索回溯的方法，也可以用DFS序查询。前者的优点在于复杂度比较优秀，后者的优点在于兼容性更强。比较明显的O(N log^2 N)的做法为对两棵树的DFS序用树套树维护起来，而O(N log N)的算法也不难想，基于搜索和权值线段树合并维护DFS序。至此，问题解决。

2.3.2 例题选讲

- [BZOJ4034] 树上操作
- 题意：
- 给出一棵N个点的树，要求支持M次操作：①给树上某个点加上一个数；②给某个点为根的子树每个节点都加上一个数；③询问某个点到根路径上的点权和。
- N, M≤100000

2.3.2 例题选讲

- [BZOJ4034] 树上操作
- 题意：
- 给出一棵N个点的树，要求支持M次操作：①给树上某个点加上一个数；②给某个点为根的子树每个节点都加上一个数；③询问某个点到根路径上的点权和。
- N, M≤100000

- 标准算法：
- 入栈出栈序左端点加上该点点权，右端点减去该点点权，就可以消除当前区间对其他询问（DFS序包含当前DFS区间但是询问并不包含当前区间）的影响。线段树中，每个点多维护一个信息，为区间内DFS序左端点的个数减DFS序右端点的个数，方便区间加标记下传。查询时只需要返回前缀和即可。

2.4 树链剖分

2.4.1 方法概述

- 为了实现链修改，可以调整DFS遍历树的过程，使得任意点到根的链能被分为log N个DFS序区间。

2.4.1 方法概述

- 为了实现链修改，可以调整DFS遍历树的过程，使得任意点到根的链能被分为log N个DFS序区间。

- 保证父亲与子树点数最大的儿子DFS序连续的树链剖分称为重链剖分，保证父亲与深度最大的儿子DFS序连续的树链剖分称为长链剖分。

2.4.1 方法概述

- 为了实现链修改，可以调整DFS遍历树的过程，使得任意点到根的链能被分为log N个DFS序区间。
- 保证父亲与子树点数最大的儿子DFS序连续的树链剖分称为重链剖分，保证父亲与深度最大的儿子DFS序连续的树链剖分称为长链剖分。
- 树链剖分使用的一般条件：
- 1、信息可合并；
- 2、树的形态修改很少或没有。
- 通常情况下树链剖分是解决树上问题的有力手段，但是复杂度一般较大。

2.4.2 例题选讲

- [Codeforces600E] Lomsat gelral
- 题意：
- 给出一棵N个点的树，求其所有子树内出现次数最多的颜色编号和。
- N≤100000

2.4.2 例题选讲

- [Codeforces600E] Lomsat gelral
- 题意：
- 给出一棵N个点的树，求其所有子树内出现次数最多的颜色编号和。
- N≤100000

- 标准算法：
- 对整棵树进行重链剖分。
- 然后DFS统计答案。对于每个节点，按照依次DFS轻儿子、清空轻儿子标记——遍历重儿子——重新计算轻儿子标记并与重儿子标记合并——计算当前点答案来处理。

2.4.2 例题选讲

- [Codeforces600E] Lomsat gelral
- 题意：
- 给出一棵N个点的树，求其所有子树内出现次数最多的颜色编号和。
- N≤100000

- 标准算法：
- 对整棵树进行重链剖分。
- 然后DFS统计答案。对于每个节点，按照依次DFS轻儿子、清空轻儿子标记——遍历重儿子——重新计算轻儿子标记并与重儿子标记合并——计算当前点答案来处理。
- 看起来复杂度是O(N^2)？

2.4.2 例题选讲

- [Codeforces600E] Lomsat gelral
- 题意：
- 给出一棵N个点的树，求其所有子树内出现次数最多的颜色编号和。
- N≤100000

- 标准算法：
- 对整棵树进行重链剖分。
- 然后DFS统计答案。对于每个节点，按照依次DFS轻儿子、清空轻儿子标记——遍历重儿子——重新计算轻儿子标记并与重儿子标记合并——计算当前点答案来处理。
- 看起来复杂度是O(N^2)？
- 注意了，我们的做法是基于重链剖分的。因为每个点到根路径上的轻边不会超过log条，所以每个点作为轻儿子最多只会被重复计算log次。时间复杂度O(N log N)。

2.4.2 例题选讲

- [BZOJ3626] LCA
- 题意：
- 给出一棵N个点的树，要求支持Q次询问，每次询问一个点z与编号为区间[l,r]内的点分别求最近公共祖先得到的最近公共祖先深度和。
- N, Q≤50000

2.4.2 例题选讲

- [BZOJ3626] LCA
- 题意：
- 给出一棵N个点的树，要求支持Q次询问，每次询问一个点z与编号为区间[l,r]内的点分别求最近公共祖先得到的最近公共祖先深度和。
- N, Q≤50000
- 标准算法：
- 考虑这样的一种暴力，我们把z到根上的点全部打标记，对于l到r之间的点，向上搜索到第一个有标记的点求出它的深度统计答案。观察到，深度其实就是上面有几个已标记了的点（包括自身）。所以，我们不妨把z到根的路径上的点全部+1，对于l到r之间的点询问他们到根路径上的点权和。仔细观察不难发现，这个操作具有叠加性，且可逆。也就是说我们可以对于l到r之间的点i，将i到根的路径上的点全部+1， 转而询问z到根的路径上的点（包括自身）的权值和就是这个询问的答案。把询问差分一下，也就是用[1,r]-[1,l-1]来计算答案。从0到n-1依次插入点i，即将i到根的路径上的点全部+1，离线回答询问即可。

2.5 动态树

2.5.1 方法概述

- 动态树是一种强大的数据结构，相当于用平衡树维护的动态的树链剖分。

2.5.1 方法概述

- 动态树是一种强大的数据结构，相当于用平衡树维护的动态的树链剖分。

- 动态树使用的一般场合：
- 1、信息可合并；
- 2、子树修改很少或没有。

2.5.1 方法概述

- 动态树是一种强大的数据结构，相当于用平衡树维护的动态的树链剖分。

- 动态树使用的一般场合：
- 1、信息可合并；
- 2、子树修改很少或没有。

- 另外，因为动态树代码难度和常数因子均较大，对于比较简单或对常数要求比较高的题目不建议使用。

2.5.2 例题选讲

- [BZOJ2959] 长跑
- 题意：
- 给出一张N个点没有边的无向图，要求支持M次操作：
- ①加入一条边；
- ②修改点权；
- ③求从一个点出发，对于所有的无向边都只沿着一个方向走，到达另一个点，路径所覆盖的最大点权和。
- $N \le 150000, M \le 750000$

2.5.2 例题选讲

- [BZOJ2959] 长跑
- 题意：
- 给出一张N个点没有边的无向图，要求支持M次操作：
- ①加入一条边；
- ②修改点权；
- ③求从一个点出发，对于所有的无向边都只沿着一个方向走，到达另一个点，路径所覆盖的最大点权和。
- N≤150000, M≤750000
- 标准算法：
- 题意即为动态维护边双连通分量，每次加边可能会把一条链缩起来，每次询问即为在缩点后树中的链上点权和。
- 考虑边双连通分量的合并，不妨用并查集来维护。但是，这是不是意味着时时处处都要getfather，感觉整个LCT都乱了？其实不然，只需要在Access的时候getfather即可。

2.5.2 例题选讲

- [Codeforces19E] Fairy
- 题意：
- 给出一张N个点M条边的无向图，对于每一条边分别询问删去该边后全图是否为二分图。其中N, M≤10000。

2.5.2 例题选讲

- [Codeforces19E] Fairy
- 题意：
- 给出一张N个点M条边的无向图，对于每一条边分别询问删去该边后全图是否为二分图。其中N, M≤10000。
- 标准算法一：动态树
- 实质相当于维护动态二分图。
- 依次向图中添边。如果添加一条边之后形成了一个环，那么就删除环上删除时间最早的边；特殊地，如果这个环是奇环，删除之前要把这条边存到平衡树中。
- 删边时，那条边可能在动态树中、平衡树中，在对应位置把它删去即可。

2.5.2 例题选讲

- [Codeforces19E] Fairy
- 题意：
- 给出一张N个点M条边的无向图，对于每一条边分别询问删去该边后全图是否为二分图。其中N, M≤10000。
- 标准算法二：时间线段树+按秩合并并查集
- 按时间分治。线段树在每个节点上打上标记，表示在这个时间段上，这条边一直存在。使用按秩合并并查集判断二分图。

2.5.2 例题选讲

- [Codeforces19E] Fairy
- 题意：
- 给出一张N个点M条边的无向图，对于每一条边分别询问删去该边后全图是否为二分图。其中N, M≤10000。
- 标准算法三：深度遍历生成树+树上差分
- 首先，求图的深度遍历生成树，保证所有的非树边都是返祖边。一条返祖边在树上对应着一个环。
- 接下来进一步分析。如果一个图是二分图，那么必定不存在奇环；反之，则必定存在奇环。要破坏所有的奇环，等价于求所有奇环边集的交。
- 但是，如果两条返祖边的环有交集，那么这两个环必然互相影响。可以知道：如果这两个环的奇偶性相同，那么最终必然形成偶环；如果这两个环的奇偶性不同，那么最终必定形成奇环。所以，选出的边，还不应该包含任何的偶环。
- 判每条边和环的关系可以用树上差分来实现。至此问题解决。

2.6 点分树

2.6.1 方法概述

- 对于一些带有单点修改或者范围修改的问题，可以考虑把点分治得到的各层重心建成点分树。点分树的树高是log N级别的。

2.6.1 方法概述

- 对于一些带有单点修改或者范围修改的问题，可以考虑把点分治得到的各层重心建成点分树。点分树的树高是log N级别的。

- 点分树使用的一般条件：
- 1、修改操作与树的形态关联不大；
- 2、询问全树链信息。

2.6.2 例题选讲

- [BZOJ1095] 捉迷藏
- 题意：
- 给出一棵N个点的树，点分为黑点和白点，边不带权，要求支持M次操作：①把一个黑点变为白点，或者把一个白点变为黑点；②询问最远黑点对的距离。
- $N \le 100000$, $M \le 500000$

2.6.2 例题选讲

- [BZOJ1095] 捉迷藏
- 题意：
- 给出一棵N个点的树，点分为黑点和白点，边不带权，要求支持M次操作：①把一个黑点变为白点，或者把一个白点变为黑点；②询问最远黑点对的距离。
- N≤100000, M≤500000

- 标准算法一：
- 建立点分树。每个点上维护两个堆：①点分树上这个点对应子树中所有黑点到点分树上这个点父亲的最远距离；②点分树上这个点直接儿子的第①个堆堆顶最大值。在此基础上再维护一个全局距离最大值的堆即可。

2.6.2 例题选讲

- [BZOJ1095] 捉迷藏
- 题意：
- 给出一棵N个点的树，点分为黑点和白点，边不带权，要求支持M次操作：①把一个黑点变为白点，或者把一个白点变为黑点；②询问最远黑点对的距离。
- N≤100000, M≤500000

- 标准算法二：
- 线段树维护括号序列。

2.7 树上莫队

2.7.1 方法概述

- 树上莫队类似普通莫队。普通莫队基于分块，树上莫队基于树分块；普通莫队指针在序列上一个一个扫元素，树上莫队指针在树上一个一个爬节点。

2.7.1 方法概述

- 树上莫队类似普通莫队。普通莫队基于分块，树上莫队基于树分块；普通莫队指针在序列上一个一个扫元素，树上莫队指针在树上一个一个爬节点。
- 对于修改操作，可以类比带修改莫队设计带修改树上莫队。

2.7.1 方法概述

- 树上莫队类似普通莫队。普通莫队基于分块，树上莫队基于树分块；普通莫队指针在序列上一个一个扫元素，树上莫队指针在树上一个一个爬节点。
- 对于修改操作，可以类比带修改莫队设计带修改树上莫队。
- 树上莫队使用的一般条件：
- 支持离线。

2.7.1 方法概述

- 树上莫队类似普通莫队。普通莫队基于分块，树上莫队基于树分块；普通莫队指针在序列上一个一个扫元素，树上莫队指针在树上一个一个爬节点。
- 对于修改操作，可以类比带修改莫队设计带修改树上莫队。
- 树上莫队使用的一般条件：
- 支持离线。
- 树上莫队虽然理论复杂度并不优秀，但是常数非常小，有时候可能取得意想不到的效果。

2.7.2 例题选讲

- [BZOJ1146] 网络管理
- 题意：
- 给出一棵N个点的树，M次操作，带修改求链上第K大。
- N≤80000, M≤80000

2.7.2 例题选讲

- [BZOJ1146] 网络管理
- 题意：
- 给出一棵N个点的树，M次操作，带修改求链上第K大。
- N≤80000, M≤80000

- 标准算法：
- 带修改树上莫队。

2 更多练习（中等）

- [BZOJ2002] 弹飞绵羊
- [BZOJ2325] 道馆之战
- [BZOJ3083] 遥远的国度
- [BZOJ3531] 旅行
- [BZOJ3669] 魔法森林
- [BZOJ3730] 震波
- [BZOJ4025] 二分图
- [BZOJ4127] Abs
- [BZOJ4196] 软件包管理器

2 更多练习（困难）

- [BZOJ1758] 重建计划
- [BZOJ3052] 糖果公园
- [BZOJ3653] 谈笑风生
- [BZOJ3772] 精神污染
- [BZOJ3784] 树上的路径
- [BZOJ3924] 幻想乡战略游戏
- [Codeforces482E] ELCA
- [Codeforces603E] Pastoral Oddities
- [Codeforces786E] ALT

3 树的计数

目录

- 3.1 有向无环图与连通块计数
- 3.2 prufer序列与无根树计数
- 3.3 矩阵树定理与生成树计数

3.1 有向无环图与连通块计数

- 对于一类树上连通块相关的计数问题，可以考虑在点分治的基础上按照树的DFS序建图，L[i]向R[i]+1连边表示不选这个点，L[i]向L[i]+1连边表示选这个点，将其转化为有向无环图上的问题。

3.2 prufer序列与无根树计数

- prufer序列：对于一棵带标号无根树，依次写下其编号最小的叶子的父亲编号，之后删去这个叶子，直到只剩下两个点。可以发现，prufer序列与带标号无根树的形态一一对应。也就是说，N个节点的带标号无根树形态共有N^(N-2)种。

3.2 prufer序列与无根树计数

- prufer序列：对于一棵带标号无根树，依次写下其编号最小的叶子的父亲编号，之后删去这个叶子，直到只剩下两个点。可以发现，prufer序列与带标号无根树的形态一一对应。也就是说，N个节点的带标号无根树形态共有N^(N-2)种。

- 关于prufer序列还有一个推论：把点数分别为a1、a2、a3、……、an的连通块连成树的方案数为(∏a)(∑a)^(n-2)。这个可以类似prufer序列的生成方法来证明。

3.3 矩阵树定理与生成树计数

- 基尔霍夫矩阵第i行第i列元素的值为第i个点在图中的度，第i行第j列元素的值为图中连接第i个点和第j个点边数的相反数。

3.3 矩阵树定理与生成树计数

- 基尔霍夫矩阵第i行第i列元素的值为第i个点在图中的度，第i行第j列元素的值为图中连接第i个点和第j个点边数的相反数。
- 矩阵树定理：基尔霍夫矩阵的任何一个n-1阶主余子式的值即为原图的生成树个数。

3 更多练习（困难）

- 最普遍的方法还是组合计数。

- [BZOJ4011] 落忆枫音
- [Codeforces599E] Sandy and Nuts
- [Codeforces762F] Tree nesting
- [Codeforces724F] Uniformly Branched Trees
- [Codeforces914H] Ember and Storm's Tree Game

4 无向图的生成树

目录

- 4.1 最小生成树
- 4.2 最短路径树
- 4.3 Tarjan生成树
- 4.4 Kruskal生成树
- 4.5 DFS遍历生成树
- 4.6 BFS遍历生成树

4.1 最小生成树

- 最小生成树是图边权和最小的生成树。

4.1 最小生成树

- 最小生成树是图边权和最小的生成树。
- 最小生成树上两点间路径最大边边权为图上这两点间所有路径最大边边权的最小值。

4.1 最小生成树

- 最小生成树是图边权和最小的生成树。
- 最小生成树上两点间路径最大边边权为图上这两点间所有路径最大边边权的最小值。
- 最大生成树同理。

4.2 最短路径树

- 根到某个点最短路径树上的边权和即为图上根到这个点的最短路长度。

4.2 最短路径树

- 根到某个点最短路径树上的边权和即为图上根到这个点的最短路长度。
- 一般考虑用Dijkstra算法建树。

4.3 Tarjan生成树

- 用Tarjan把点双连通分量缩成点可以构建圆方树。

4.3 Tarjan生成树

- 用Tarjan把点双连通分量缩成点可以构建圆方树。
- 圆方树中的方点代表点双连通分量，方点之间的圆点即为割点。

4.3 Tarjan生成树

- 用Tarjan把点双连通分量缩成点可以构建圆方树。
- 圆方树中的方点代表点双连通分量，方点之间的圆点即为割点。
- 可以方便地进行点双连通分量的信息维护。
- 特别适用于仙人掌相关问题。

4.3 Tarjan生成树

- 用Tarjan把点双连通分量缩成点可以构建圆方树。
- 圆方树中的方点代表点双连通分量，方点之间的圆点即为割点。
- 可以方便地进行点双连通分量的信息维护。
- 特别适用于仙人掌相关问题。
- 用Tarjan把边双连通分量缩成点同样可以构树。

4.3 Tarjan生成树

- 用Tarjan把点双连通分量缩成点可以构建圆方树。
- 圆方树中的方点代表点双连通分量，方点之间的圆点即为割点。
- 可以方便地进行点双连通分量的信息维护。
- 特别适用于仙人掌相关问题。
- 用Tarjan把边双连通分量缩成点同样可以构树。
- 树中的每个点代表边双连通分量，每条边都是桥边。

4.4 Kruskal生成树

- 在Kruskal算法的基础上把最小生成树的边建成点，在合并点集的同时对边建成的点连边。

4.4 Kruskal生成树

- 在Kruskal算法的基础上把最小生成树的边建成点，在合并点集的同时对边建成的点连边。
- Kruskal生成树从下往上点权递增。

4.4 Kruskal生成树

- 在Kruskal算法的基础上把最小生成树的边建成点，在合并点集的同时对边建成的点连边。
- Kruskal生成树从下往上点权递增。
- 图上边权不超过某特定常数的边形成的连通块能够对应Kruskal生成树上的子树。

4.5 DFS生成树

- DFS生成树没有横叉边。

4.5 DFS生成树

- DFS生成树没有横叉边。
- 可用于求解特殊的环问题。

4.6 BFS生成树

- BFS生成树没有返祖边。

4.6 BFS生成树

- BFS生成树没有返祖边。
- 适用于某些动态规划对树的计数。

4 更多练习（中等）

- [BZOJ1576] Safe Travel
- [BZOJ2001] 城市建设
- [BZOJ2654] Tree
- [BZOJ3206] 道路费用
- [BZOJ3440] 传球游戏
- [BZOJ3551] Peaks加强版
- [BZOJ3714] Kuglarz
- [Codeforces632F] Magic Matrix
- [Codeforces814E] An unavoidable detour for home
- [Codeforces715B] Complete The Graph
- [Codeforces487E] Tourists

谢谢观看

By zhan8855