# 树分治的应用

Claris

Hangzhou Dianzi University

2017 年 7 月 25 日

# Overview

- 树分治分为点分治和边分治，思想是选出一个点或者一条边，计算经过它的答案，然后将树分裂成若干子树递归分治解决。

# Overview

- 树分治分为点分治和边分治，思想是选出一个点或者一条边，计算经过它的答案，然后将树分裂成若干子树递归分治解决。
- 当选择的点是树的重心时，剩余每棵子树的大小不超过当前树的一半。

# Overview

- 树分治分为点分治和边分治，思想是选出一个点或者一条边，计算经过它的答案，然后将树分裂成若干子树递归分治解决。
- 当选择的点是树的重心时，剩余每棵子树的大小不超过当前树的一半。
- 故一直递归下去的总时间复杂度为 $O(n\log n)$。

# Overview

- 树分治分为点分治和边分治，思想是选出一个点或者一条边，计算经过它的答案，然后将树分裂成若干子树递归分治解决。
- 当选择的点是树的重心时，剩余每棵子树的大小不超过当前树的一半。
- 故一直递归下去的总时间复杂度为 $O(n \log n)$。
- 边分治则需要通过加虚点来限制度数，否则会退化到平方级别的复杂度。

# Tree

给定一棵 $n$ 个点的树，求距离不超过 $k$ 的点对数。

# Tree

给定一棵 $n$ 个点的树，求距离不超过 $k$ 的点对数。

- $1 \leq n \leq 40000$。

# Tree

给定一棵 $n$ 个点的树，求距离不超过 $k$ 的点对数。

- $1 \leq n \leq 40000$。
- $0 \leq k \leq 10^9$。

## Tree

给定一棵 $n$ 个点的树，求距离不超过 $k$ 的点对数。

- $1 \leq n \leq 40000$。
- $0 \leq k \leq 10^9$。
- Source： POJ 1741

# Solution

- 对树进行点分治，只需考虑经过重心的路径。

# Solution

- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么若 $x, y$ 属于重心下面的不同子树，且 $d_x + d_y \leq k$，则可行。

# Solution

- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么若 $x, y$ 属于重心下面的不同子树，且 $d_x + d_y \leq k$，则可行。
- 先不考虑不同子树的条件，那么只要将所有 $d$ 排序然后双指针即可完成统计。

# Solution

- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么若 $x, y$ 属于重心下面的不同子树，且 $d_x + d_y \leq k$，则可行。
- 先不考虑不同子树的条件，那么只要将所有 $d$ 排序然后双指针即可完成统计。
- 在刚才的过程中，多统计的一定是同一棵子树的贡献，对于每棵子树单独再求一次答案即可。

# Solution

- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么若 $x, y$ 属于重心下面的不同子树，且 $d_x + d_y \leq k$，则可行。
- 先不考虑不同子树的条件，那么只要将所有 $d$ 排序然后双指针即可完成统计。
- 在刚才的过程中，多统计的一定是同一棵子树的贡献，对于每棵子树单独再求一次答案即可。
- 时间复杂度 $O(n \log^2 n)$。

# Yin and Yang

给定一棵 $n$ 个点的树，每个点要么是阴性，要么是阳性。

# Yin and Yang

给定一棵 $n$ 个点的树，每个点要么是阴性，要么是阳性。

若一条路径满足阴阳平衡，则表示路径上阴性点数与阳性点数相等。

# Yin and Yang

给定一棵 $n$ 个点的树，每个点要么是阴性，要么是阳性。

若一条路径满足阴阳平衡，则表示路径上阴性点数与阳性点数相等。

求有多少条阴阳平衡的路径 $(u, v)$，且上面存在一个点 $x(x \neq u, x \neq v)$，满足 $(u, x), (x, v)$ 也是阴阳平衡的。

# Yin and Yang

给定一棵 $n$ 个点的树，每个点要么是阴性，要么是阳性。

若一条路径满足阴阳平衡，则表示路径上阴性点数与阳性点数相等。

求有多少条阴阳平衡的路径 $(u, v)$，且上面存在一个点 $x(x \neq u, x \neq v)$，满足 $(u, x), (x, v)$ 也是阴阳平衡的。

- $1 \leq n \leq 100000$。

## Yin and Yang

给定一棵 $n$ 个点的树，每个点要么是阴性，要么是阳性。

若一条路径满足阴阳平衡，则表示路径上阴性点数与阳性点数相等。

求有多少条阴阳平衡的路径 $(u, v)$，且上面存在一个点 $x(x \neq u, x \neq v)$，满足 $(u, x), (x, v)$ 也是阴阳平衡的。

- $1 \leq n \leq 100000$。
- Source：USACO 2013 Open

## Solution

- 将阴看成 -1，阳看成 1，那么若一条路径上的点权之和为 0，则是阴阳平衡的。

# Solution

- 将阴看成 -1，阳看成 1，那么若一条路径上的点权之和为 0，则是阴阳平衡的。
- 对树进行点分治，只需考虑经过重心的路径。

# Solution

- 将阴看成 -1，阳看成 1，那么若一条路径上的点权之和为 0，则是阴阳平衡的。
- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么 $x, y$ 到重心路径上，$d_x$ 与 $d_y$ 至少有一个出现了两次以上。

# Solution

- 将阴看成 -1，阳看成 1，那么若一条路径上的点权之和为 0，则是阴阳平衡的。
- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么 $x, y$ 到重心路径上，$d_x$ 与 $d_y$ 至少有一个出现了两次以上。
- 依次遍历每棵子树，枚举每个点作为 $x$，按 $d$ 出现 1 次或多次分类统计贡献，然后将这棵子树加入 $y$ 集合。

# Solution

- 将阴看成 -1，阳看成 1，那么若一条路径上的点权之和为 0，则是阴阳平衡的。
- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的距离为 $d_x$，那么 $x, y$ 到重心路径上，$d_x$ 与 $d_y$ 至少有一个出现了两次以上。
- 依次遍历每棵子树，枚举每个点作为 $x$，按 $d$ 出现 1 次或多次分类统计贡献，然后将这棵子树加入 $y$ 集合。
- 时间复杂度 $O(n \log n)$。

# 重建计划

给定一棵 $n$ 个点的树，找一条边数在 $[L, R]$ 之间的路径，使得边权平均数最大。

# 重建计划

给定一棵 $n$ 个点的树，找一条边数在 $[L, R]$ 之间的路径，使得边权平均数最大。

- $1 \le n \le 100000$。

# 重建计划

给定一棵 $n$ 个点的树，找一条边数在 $[L, R]$ 之间的路径，使得边权平均数最大。

- $1 \leq n \leq 100000$。
- Source： WC 2010

# Solution

- 经典 0/1 分数规划模型，首先二分答案，将每条边权值都减去 $mid$，若存在一条边数在 $[L, R]$ 之间的权值和非负的路径，则 $ans \geq mid$。

# Solution

- 经典 0/1 分数规划模型，首先二分答案，将每条边权值都减去 $mid$，若存在一条边数在 $[L, R]$ 之间的权值和非负的路径，则 $ans \geq mid$。
- 对树进行点分治，只需考虑经过重心的路径。

## Solution

- 经典 0/1 分数规划模型，首先二分答案，将每条边权值都减去 $mid$，若存在一条边数在 $[L, R]$ 之间的权值和非负的路径，则 $ans \geq mid$。
- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的边数为 $d_x$，权值和为 $w_x$，则 $L \leq d_x + d_y \leq R$。

# Solution

- 经典 0/1 分数规划模型，首先二分答案，将每条边权值都减去 $mid$，若存在一条边数在 $[L, R]$ 之间的权值和非负的路径，则 $ans \geq mid$。
- 对树进行点分治，只需考虑经过重心的路径。
- 设 $x$ 到重心的边数为 $d_x$，权值和为 $w_x$，则 $L \leq d_x + d_y \leq R$。
- 依次遍历每棵子树，枚举每个点作为 $x$，则要找到最大的 $w_y$ 满足 $L - d_x \leq d_y \leq R - d_x$，然后将这棵子树加入 $y$ 集合。

# Solution

- 对于 $y$ 集合中的点，按 $d$ 从小到大排序，那么若按 $d$ 从小到大枚举 $x$，则对应 $y$ 集合中滑动窗口的最大值，单调队列即可解决。

# Solution

- 对于 $y$ 集合中的点，按 $d$ 从小到大排序，那么若按 $d$ 从小到大枚举 $x$，则对应 $y$ 集合中滑动窗口的最大值，单调队列即可解决。
- 从重心开始 BFS 即可在线性时间内完成排序。

## Solution

- 对于 $y$ 集合中的点，按 $d$ 从小到大排序，那么若按 $d$ 从小到大枚举 $x$，则对应 $y$ 集合中滑动窗口的最大值，单调队列即可解决。
- 从重心开始 BFS 即可在线性时间内完成排序。
- 为了使得复杂度只和子树大小有关，按子树最大深度从小到大考虑每棵子树即可。

# Solution

- 对于 $y$ 集合中的点，按 $d$ 从小到大排序，那么若按 $d$ 从小到大枚举 $x$，则对应 $y$ 集合中滑动窗口的最大值，单调队列即可解决。
- 从重心开始 BFS 即可在线性时间内完成排序。
- 为了使得复杂度只和子树大小有关，按子树最大深度从小到大考虑每棵子树即可。
- 时间复杂度 $O(n \log n \log w)$。

## 树上的回文串

给定一棵 $n$ 个点的树，每条边上有一个小写字符。

# 树上的回文串

给定一棵 $n$ 个点的树，每条边上有一个小写字符。

找一条边数最多的路径，使得起点到终点经过的边上的字符形成一个回文串。

# 树上的回文串

给定一棵 $n$ 个点的树，每条边上有一个小写字符。

找一条边数最多的路径，使得起点到终点经过的边上的字符形成一个回文串。

- $2 \le n \le 50000$。

## Solution

- 显然奇回文串和偶回文串互不影响，枚举长度的奇偶性然后二分答案，转化为判定是否存在长度为 $K$ 的回文串。

# Solution

- 显然奇回文串和偶回文串互不影响，枚举长度的奇偶性然后二分答案，转化为判定是否存在长度为 $K$ 的回文串。
- 对树进行点分治，设到重心长度分别为 $d_x$ 和 $d_y$，那么回文中心必然在 $d$ 较大的那边，不妨设 $d_x \geq d_y$。

# Solution

- 显然奇回文串和偶回文串互不影响，枚举长度的奇偶性然后二分答案，转化为判定是否存在长度为 $K$ 的回文串。
- 对树进行点分治，设到重心长度分别为 $d_x$ 和 $d_y$，那么回文中心必然在 $d$ 较大的那边，不妨设 $d_x \geq d_y$。
- 根据重心到 $x$ 路径的字符串以及回文中心的位置，可以立即得到重心到 $y$ 的路径应该是什么串，利用字符串 Hash 可以转化为判定是否存在一个合法的 $y$ 满足与 $x$ 不属于同一棵子树。

# Solution

- 将所有 Hash 值插入散列表，对于每种值维护它在哪些子树出现过，若有多个，保留任意 2 个。

# Solution

- 将所有 Hash 值插入散列表，对于每种值维护它在哪些子树出现过，若有多个，保留任意 2 个。
- 那么在那 2 个子树中，必有一个和 $x$ 不是同一子树。

## Solution

- 将所有 Hash 值插入散列表，对于每种值维护它在哪些子树出现过，若有多个，保留任意 2 个。
- 那么在那 2 个子树中，必有一个和 $x$ 不是同一子树。
- 时间复杂度 $O(n\log^2 n)$。

# 寝室管理

给定一张 $n$ 个点，$n$ 条边的无向连通图，求有多少条经过点数不超过 $k$ 的简单路径。

# 寝室管理

给定一张 $n$ 个点，$n$ 条边的无向连通图，求有多少条经过点数不超过 $k$ 的简单路径。

- $2 \leq k \leq n \leq 100000$。

# 寝室管理

给定一张 $n$ 个点，$n$ 条边的无向连通图，求有多少条经过点数不超过 $k$ 的简单路径。

- $2 \leq k \leq n \leq 100000$。
- Source：BZOJ 3648

# Solution

- 若去掉一条多余的边 $(a, b)$，则剩下的图是棵树。

# Solution

- 若去掉一条多余的边 $(a, b)$，则剩下的图是棵树。
- 对树直接进行点分治，即可求出树上长度不超过 $k$ 的路径数。

# Solution

- 若去掉一条多余的边 $(a, b)$，则剩下的图是棵树。
- 对树直接进行点分治，即可求出树上长度不超过 $k$ 的路径数。
- 考虑 $(a, b)$ 的贡献，一定是从某个点 $x$ 出发到达 $a$，再经过 $(a, b)$，然后再到达某个点 $y$。

# Solution

- 在树上找到 $a$ 到 $b$ 的路径，称为主干。

# Solution

- 在树上找到 $a$ 到 $b$ 的路径，称为主干。
- 那么主干上一定存在一条分界线，前面的点 (包括下面子树) 走到 $a$，剩余的点走到 $b$。

# Solution

- 在树上找到 $a$ 到 $b$ 的路径，称为主干。
- 那么主干上一定存在一条分界线，前面的点 (包括下面子树) 走到 $a$，剩余的点走到 $b$。
- 枚举分界线，用树状数组完成统计即可。

# Solution

- 在树上找到 $a$ 到 $b$ 的路径，称为主干。
- 那么主干上一定存在一条分界线，前面的点 (包括下面子树) 走到 $a$，剩余的点走到 $b$。
- 枚举分界线，用树状数组完成统计即可。
- 时间复杂度 $O(n\log^2 n)$。

# Summary

- 前面的问题都是直接一次点分治就可以解决。

# Summary

- 前面的问题都是直接一次点分治就可以解决。
- 将点分治的过程记下来，我们会得到一个分治结构，每个点只会被 $O(\log n)$ 个重心管辖。

## Summary

- 前面的问题都是直接一次点分治就可以解决。
- 将点分治的过程记下来，我们会得到一个分治结构，每个点只会被 $O(\log n)$ 个重心管辖。
- 这使得它可以很容易配合其它数据结构进行维护。

# Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

# Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

将树上 $\frac{n(n-1)}{2}$ 条路径按照长度从大到小排序，求第 $k$ 长的路径的长度。

# Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

将树上 $\frac{n(n-1)}{2}$ 条路径按照长度从大到小排序，求第 $k$ 长的路径的长度。

- $2 \le n \le 50000$。

## Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

将树上 $\frac{n(n-1)}{2}$ 条路径按照长度从大到小排序，求第 $k$ 长的路径的长度。

- $2 \le n \le 50000$。
- $1 \le k \le \frac{n(n-2)}{2}$。

## Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

将树上 $\frac{n(n-1)}{2}$ 条路径按照长度从大到小排序，求第 $k$ 长的路径的长度。

- $2 \le n \le 50000$。
- $1 \le k \le \frac{n(n-2)}{2}$。
- $1 \le w_i \le 10000$。

# Lady CA and the graph

给定一棵 $n$ 个点的树，每条边有边权 $w_i$。

将树上 $\frac{n(n-1)}{2}$ 条路径按照长度从大到小排序，求第 $k$ 长的路径的长度。

- $2 \le n \le 50000$。
- $1 \le k \le \frac{n(n-2)}{2}$。
- $1 \le w_i \le 10000$。
- Source：HDU 5664

# Solution

- 二分答案，转化为统计有多少条长度不小于 $mid$ 的路径。

# Solution

- 二分答案，转化为统计有多少条长度不小于 $mid$ 的路径。
- 直接点分治统计的话，单次检验复杂度为 $O(n \log^2 n)$。

# Solution

- 二分答案，转化为统计有多少条长度不小于 $mid$ 的路径。
- 直接点分治统计的话，单次检验复杂度为 $O(n\log^2 n)$。
- 总复杂度 $O(n\log^2 n\log w)$，不能接受。

# Solution

- 每次树分治的过程都是一样的。

# Solution

- 每次树分治的过程都是一样的。
- 将树分治的过程记下来，并对于每个分治结构，将所有点按到重心的距离从小到大排序。

# Solution

- 每次树分治的过程都是一样的。
- 将树分治的过程记下来，并对于每个分治结构，将所有点按到重心的距离从小到大排序。
- 预处理时间复杂度 $O(n\log^2 n)$。

## Solution

- 每次树分治的过程都是一样的。
- 将树分治的过程记下来，并对于每个分治结构，将所有点按到重心的距离从小到大排序。
- 预处理时间复杂度 $O(n \log^2 n)$。
- 那么每次检验的时候，只需要双指针即可完成统计，时间复杂度 $O(n \log n)$。

# Solution

- 每次树分治的过程都是一样的。
- 将树分治的过程记下来，并对于每个分治结构，将所有点按到重心的距离从小到大排序。
- 预处理时间复杂度 $O(n\log^2 n)$。
- 那么每次检验的时候，只需要双指针即可完成统计，时间复杂度 $O(n\log n)$。
- 时间复杂度 $O(n\log n\log w)$。

## 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。

# 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。

支持 $m$ 次操作：

# 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。

支持 $m$ 次操作：

(1) 查询所有离 $x$ 距离不超过 $k$ 的点的权值之和。

## 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。

支持 $m$ 次操作：

(1) 查询所有离 $x$ 距离不超过 $k$ 的点的权值之和。

(2) 修改点 $x$ 的权值。

## 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。

支持 $m$ 次操作：

(1) 查询所有离 $x$ 距离不超过 $k$ 的点的权值之和。

(2) 修改点 $x$ 的权值。

- $1 \leq n \leq 100000$。

## 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。
支持 $m$ 次操作：
(1) 查询所有离 $x$ 距离不超过 $k$ 的点的权值之和。
(2) 修改点 $x$ 的权值。

- $1 \leq n \leq 100000$。
- $1 \leq m \leq 100000$。

# 震波

给定一棵 $n$ 个点的树，每条边长度都是 1。
支持 $m$ 次操作：
(1) 查询所有离 $x$ 距离不超过 $k$ 的点的权值之和。
(2) 修改点 $x$ 的权值。

- $1 \leq n \leq 100000$。
- $1 \leq m \leq 100000$。
- Source： BZOJ 3730

# Solution

- 如果没有修改，那么就是个简单的点分治。

## Solution

- 如果没有修改，那么就是个简单的点分治。
- 有修改，则把点分治的过程记录下来，每个分治结构按到重心的距离用树状数组维护。

# Solution

- 如果没有修改，那么就是个简单的点分治。
- 有修改，则把点分治的过程记录下来，每个分治结构按到重心的距离用树状数组维护。
- 对于每次修改操作，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行修改。

# Solution

- 如果没有修改，那么就是个简单的点分治。
- 有修改，则把点分治的过程记录下来，每个分治结构按到重心的距离用树状数组维护。
- 对于每次修改操作，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行修改。
- 对于每次查询，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行查询。

# Solution

- 如果没有修改，那么就是个简单的点分治。
- 有修改，则把点分治的过程记录下来，每个分治结构按到重心的距离用树状数组维护。
- 对于每次修改操作，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行修改。
- 对于每次查询，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行查询。
- 为了防止相同子树的贡献多算，需要再对每个子树维护树状数组来减掉。

# Solution

- 如果没有修改，那么就是个简单的点分治。
- 有修改，则把点分治的过程记录下来，每个分治结构按到重心的距离用树状数组维护。
- 对于每次修改操作，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行修改。
- 对于每次查询，枚举所有经过 $x$ 的分治结构，在对应树状数组中进行查询。
- 为了防止相同子树的贡献多算，需要再对每个子树维护树状数组来减掉。
- 总复杂度 $O(n\log^2 n)$。

# 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

# 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

## 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

## 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

(2) 修改某条边的长度。

# 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

(2) 修改某条边的长度。

(3) 查询两个黑点在树上的路径的边权和的最大值。

## 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

(2) 修改某条边的长度。

(3) 查询两个黑点在树上的路径的边权和的最大值。

- $2 \leq n, m \leq 100000$。

# 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

(2) 修改某条边的长度。

(3) 查询两个黑点在树上的路径的边权和的最大值。

- $2 \le n, m \le 100000$。
- $-500 \le w_i \le 500$。

# 蚂蚁搬家

给定一棵 $n$ 个点的树，每条边有边权 $w_i$，每个点的颜色要么为黑色，要么为白色。

支持 $m$ 次操作：

(1) 修改某个点的颜色。

(2) 修改某条边的长度。

(3) 查询两个黑点在树上的路径的边权和的最大值。

- $2 \leq n, m \leq 100000$。
- $-500 \leq w_i \leq 500$。
- Source：BZOJ 1841

## Solution

- 将点分治的结构记录下来，对于每个分治结构，维护两棵线段树。

# Solution

- 将点分治的结构记录下来，对于每个分治结构，维护两棵线段树。
- 第一棵按 DFS 序维护所有点到重心的距离，第二棵维护重心下面每个分支的最长链。

# Solution

- 将点分治的结构记录下来，对于每个分治结构，维护两棵线段树。
- 第一棵按 DFS 序维护所有点到重心的距离，第二棵维护重心下面每个分支的最长链。
- 那么当前结构对答案的贡献就是第二棵线段树的最大值 + 次大值。

# Solution

- 将点分治的结构记录下来，对于每个分治结构，维护两棵线段树。
- 第一棵按 DFS 序维护所有点到重心的距离，第二棵维护重心下面每个分支的最长链。
- 那么当前结构对答案的贡献就是第二棵线段树的最大值 + 次大值。
- 对于操作 (1)，如果是激活某个点，则直接把它距离减去 $inf$，否则加回 $inf$。

## Solution

- 将点分治的结构记录下来，对于每个分治结构，维护两棵线段树。
- 第一棵按 DFS 序维护所有点到重心的距离，第二棵维护重心下面每个分支的最长链。
- 那么当前结构对答案的贡献就是第二棵线段树的最大值 + 次大值。
- 对于操作 (1)，如果是激活某个点，则直接把它距离减去 $inf$，否则加回 $inf$。
- 对于操作 (2)，相当于子树全部加上一个值，即区间加。

# Solution

- 再开一个全局的堆来维护每个分治结构的贡献最大值即可。

# Solution

- 再开一个全局的堆来维护每个分治结构的贡献最大值即可。
- 时间复杂度 $O(n \log n + m \log^2 n)$。

## Solution

- 再开一个全局的堆来维护每个分治结构的贡献最大值即可。
- 时间复杂度 $O(n\log n + m\log^2 n)$。
- 空间复杂度为大常数的 $O(n\log n)$，不太够？

# Solution

- 再开一个全局的堆来维护每个分治结构的贡献最大值即可。
- 时间复杂度 $O(n\log n + m\log^2 n)$。
- 空间复杂度为大常数的 $O(n\log n)$，不太够？
- 注意到每个分治结构独立，离线对每个分治结构处理即可。

# Solution

- 再开一个全局的堆来维护每个分治结构的贡献最大值即可。
- 时间复杂度 $O(n\log n + m\log^2 n)$。
- 空间复杂度为大常数的 $O(n\log n)$，不太够？
- 注意到每个分治结构独立，离线对每个分治结构处理即可。
- 空间复杂度 $O(n)$。

## Summary

- 以上问题均是显式的统计维护问题，直接点分治即可解决。

# Summary

- 以上问题均是显式的统计维护问题，直接点分治即可解决。
- 其它有些问题将树分治藏得比较深，不容易直接看出是树分治。

## Summary

- 以上问题均是显式的统计维护问题，直接点分治即可解决。
- 其它有些问题将树分治藏得比较深，不容易直接看出是树分治。
- 这需要熟练掌握树分治，并懂得灵活地运用。

# 树上 01 背包

给定一棵 $n$ 个点的树，每个点有一个物品，体积为 $v_i$，价值为 $w_i$。

# 树上 01 背包

给定一棵 $n$ 个点的树，每个点有一个物品，体积为 $v_i$，价值为 $w_i$。

$m$ 次询问，每次询问 $u$ 到 $v$ 路径上用体积为 $k$ 的背包最多能背走总价值多少的物品。

# 树上 01 背包

给定一棵 $n$ 个点的树，每个点有一个物品，体积为 $v_i$，价值为 $w_i$。

$m$ 次询问，每次询问 $u$ 到 $v$ 路径上用体积为 $k$ 的背包最多能背走总价值多少的物品。

- $1 \le n \le 20000$。

# 树上 01 背包

给定一棵 $n$ 个点的树，每个点有一个物品，体积为 $v_i$，价值为 $w_i$。

$m$ 次询问，每次询问 $u$ 到 $v$ 路径上用体积为 $k$ 的背包最多能背走总价值多少的物品。

- $1 \le n \le 20000$。
- $1 \le m \le 100000$。

# 树上 01 背包

给定一棵 $n$ 个点的树，每个点有一个物品，体积为 $v_i$，价值为 $w_i$。

$m$ 次询问，每次询问 $u$ 到 $v$ 路径上用体积为 $k$ 的背包最多能背走总价值多少的物品。

- $1 \leq n \leq 20000$。
- $1 \leq m \leq 100000$。
- $1 \leq v_i, k \leq 100$。

# Solution

- 点分治解决问题的关键：将一条路径拆成两个点到重心的路径。

# Solution

- 点分治解决问题的关键：将一条路径拆成两个点到重心的路径。
- 对树进行点分治，从重心开始，对于每个点求出到重心路径上的背包数组。

# Solution

- 点分治解决问题的关键：将一条路径拆成两个点到重心的路径。
- 对树进行点分治，从重心开始，对于每个点求出到重心路径上的背包数组。
- 对于一个询问 $(u, v)$，若它们在同一个子树，则递归解决。

# Solution

- 点分治解决问题的关键：将一条路径拆成两个点到重心的路径。
- 对树进行点分治，从重心开始，对于每个点求出到重心路径上的背包数组。
- 对于一个询问 $(u, v)$，若它们在同一个子树，则递归解决。
- 否则枚举分给 $u$ 多少体积，剩余体积都分给 $v$ 即可。

## Solution

- 点分治解决问题的关键：将一条路径拆成两个点到重心的路径。
- 对树进行点分治，从重心开始，对于每个点求出到重心路径上的背包数组。
- 对于一个询问 $(u, v)$，若它们在同一个子树，则递归解决。
- 否则枚举分给 $u$ 多少体积，剩余体积都分给 $v$ 即可。
- 时间复杂度 $O(nk\log n + m\log n + mk)$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

- $2 \leq n \leq 200000$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \le l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

- $2 \le n \le 200000$。
- $0 \le p_i \le 10^6$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

- $2 \leq n \leq 200000$。
- $0 \leq p_i \leq 10^6$。
- $0 \leq q_i \leq 10^{12}$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

- $2 \leq n \leq 200000$。
- $0 \leq p_i \leq 10^6$。
- $0 \leq q_i \leq 10^{12}$。
- $0 < s_i \leq l_i \leq 2 \times 10^{11}$。

# 购票

给定一棵 $n$ 个点的以 1 为根的有根树，每条边有长度 $s_i$。

对于每个点 $i$，它一次到达某个祖先 $j$ 的代价为 $dis(i,j) \times p_i + q_i$，且要满足 $dis(i,j) \leq l_i$。

对于每个点，计算它直接或间接到达 1 号点的最小代价。

- $2 \leq n \leq 200000$。
- $0 \leq p_i \leq 10^6$。
- $0 \leq q_i \leq 10^{12}$。
- $0 < s_i \leq l_i \leq 2 \times 10^{11}$。
- Source：NOI 2014

# Solution

- 设 $d_i$ 表示 $i$ 到 1 的距离，$f_i$ 表示 $i$ 到 1 的最小代价，则：

## Solution

- 设 $d_i$ 表示 $i$ 到 1 的距离，$f_i$ 表示 $i$ 到 1 的最小代价，则：
- $f_i = q_i + \min(f_j + (d_i - d_j)p_i) = q_i + d_i p_i + \min(-d_j p_i + f_j)$，其中 $j$ 是 $i$ 的祖先，且 $d_i - d_j \le l_i$。

# Solution

- 设 $d_i$ 表示 $i$ 到 1 的距离，$f_i$ 表示 $i$ 到 1 的最小代价，则：
- $f_i = q_i + \min(f_j + (d_i - d_j)p_i) = q_i + d_ip_i + \min(-d_jp_i + f_j)$，其中 $j$ 是 $i$ 的祖先，且 $d_i - d_j \leq l_i$。
- 若是序列，则是经典斜率优化 DP 模型。

## Solution

- 设 $d_i$ 表示 $i$ 到 1 的距离，$f_i$ 表示 $i$ 到 1 的最小代价，则：
- $f_i = q_i + \min(f_j + (d_i - d_j)p_i) = q_i + d_ip_i + \min(-d_jp_i + f_j)$，其中 $j$ 是 $i$ 的祖先，且 $d_i - d_j \le l_i$。
- 若是序列，则是经典斜率优化 DP 模型。
- 对于树的情况，则比较复杂，可以使用可持久化平衡树维护凸壳来解决，但那样过于难写。

## Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。

# Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。
- 若 $x = root$，则用 $x$ 更新树中所有点，然后递归分治。

# Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。
- 若 $x = root$，则用 $x$ 更新树中所有点，然后递归分治。
- 否则，优先递归分治 $x$ 所在子树，然后将剩下的点按 $d - l$ 从大到小排序。

## Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。
- 若 $x = root$，则用 $x$ 更新树中所有点，然后递归分治。
- 否则，优先递归分治 $x$ 所在子树，然后将剩下的点按 $d - l$ 从大到小排序。
- 对 $root$ 到 $x$ 路径上所有点维护凸壳，按照 $d - l$ 的限制依次加入每条直线。

## Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。
- 若 $x = root$，则用 $x$ 更新树中所有点，然后递归分治。
- 否则，优先递归分治 $x$ 所在子树，然后将剩下的点按 $d - l$ 从大到小排序。
- 对 $root$ 到 $x$ 路径上所有点维护凸壳，按照 $d - l$ 的限制依次加入每条直线。
- 对于每个询问，在凸壳上二分查找即可，最后再递归分治其它子树。

# Solution

- 将树进行点分治，设当前的树根为 $x$，重心为 $root$。
- 若 $x = root$，则用 $x$ 更新树中所有点，然后递归分治。
- 否则，优先递归分治 $x$ 所在子树，然后将剩下的点按 $d - l$ 从大到小排序。
- 对 $root$ 到 $x$ 路径上所有点维护凸壳，按照 $d - l$ 的限制依次加入每条直线。
- 对于每个询问，在凸壳上二分查找即可，最后再递归分治其它子树。
- 时间复杂度 $O(n \log^2 n)$。

# Distance on Triangulation

给定一个凸 $n$ 边形和它的一个三角剖分，$q$ 次询问：

# Distance on Triangulation

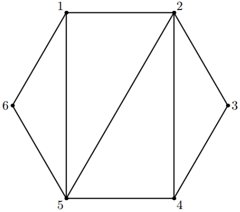
给定一个凸 $n$ 边形和它的一个三角剖分，$q$ 次询问：

每次询问给定 $u, v$，问只准经过剖分边或者多边形的边时，从 $u$ 到 $v$ 最少需要经过几条边。

# Distance on Triangulation

给定一个凸 $n$ 边形和它的一个三角剖分，$q$ 次询问：

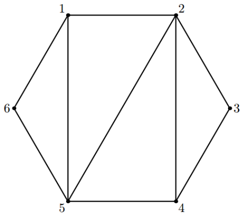
每次询问给定 $u, v$，问只准经过剖分边或者多边形的边时，从 $u$ 到 $v$ 最少需要经过几条边。



# Distance on Triangulation

给定一个凸 $n$ 边形和它的一个三角剖分，$q$ 次询问：

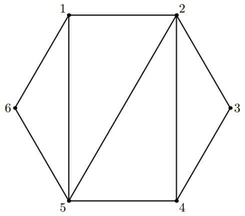
每次询问给定 $u, v$，问只准经过剖分边或者多边形的边时，从 $u$ 到 $v$ 最少需要经过几条边。




- $4 \leq n \leq 50000, 1 \leq q \leq 100000$。

# Distance on Triangulation

给定一个凸 $n$ 边形和它的一个三角剖分，$q$ 次询问：每次询问给定 $u, v$，问只准经过剖分边或者多边形的边时，从 $u$ 到 $v$ 最少需要经过几条边。




- $4 \leq n \leq 50000, 1 \leq q \leq 100000$。
- Source：NEERC 2015

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。
- 将三角形视作点，相邻三角形连边，根据三角剖分的性质，会得到一棵树。

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。
- 将三角形视作点，相邻三角形连边，根据三角剖分的性质，会得到一棵树。
- 对这棵树进行点分治，每次 DFS 求出原图中所有对应的点，然后从重心三角形的 3 个顶点分别做一次 BFS。

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。
- 将三角形视作点，相邻三角形连边，根据三角剖分的性质，会得到一棵树。
- 对这棵树进行点分治，每次 DFS 求出原图中所有对应的点，然后从重心三角形的 3 个顶点分别做一次 BFS。
- 对于每个询问，若 $u, v$ 位于同一侧，那么递归分治。

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。
- 将三角形视作点，相邻三角形连边，根据三角剖分的性质，会得到一棵树。
- 对这棵树进行点分治，每次 DFS 求出原图中所有对应的点，然后从重心三角形的 3 个顶点分别做一次 BFS。
- 对于每个询问，若 $u, v$ 位于同一侧，那么递归分治。
- 否则路径必然经过重心的 3 个顶点之一，枚举顶点更新答案即可。

# Solution

- 对图进行拓扑排序，每次取出度数为 2 的点，这样可以把所有三角形都找到。
- 将三角形视作点，相邻三角形连边，根据三角剖分的性质，会得到一棵树。
- 对这棵树进行点分治，每次 DFS 求出原图中所有对应的点，然后从重心三角形的 3 个顶点分别做一次 BFS。
- 对于每个询问，若 $u, v$ 位于同一侧，那么递归分治。
- 否则路径必然经过重心的 3 个顶点之一，枚举顶点更新答案即可。
- 时间复杂度 $O((n+q)\log n)$。

## Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

# Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

你手头有 $m$ 块钱，且你要保证你买过的点在树上互相连通，问买到的物品的总价值最多是多少。

## Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

你手头有 $m$ 块钱，且你要保证你买过的点在树上互相连通，问买到的物品的总价值最多是多少。

- $1 \leq n \leq 500$。

## Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

你手头有 $m$ 块钱，且你要保证你买过的点在树上互相连通，问买到的物品的总价值最多是多少。

- $1 \leq n \leq 500$。
- $1 \leq m \leq 4000$。

# Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

你手头有 $m$ 块钱，且你要保证你买过的点在树上互相连通，问买到的物品的总价值最多是多少。

- $1 \leq n \leq 500$。
- $1 \leq m \leq 4000$。
- $1 \leq d_i \leq 100$。

# Shopping

给定一棵有 $n$ 个点的树，第 $i$ 个点有 $d_i$ 件商品，价格为 $c_i$，价值为 $w_i$。

你手头有 $m$ 块钱，且你要保证你买过的点在树上互相连通，问买到的物品的总价值最多是多少。

- $1 \le n \le 500$。
- $1 \le m \le 4000$。
- $1 \le d_i \le 100$。
- Source：BZOJ 4182

# Solution

- 考虑某个点必选的情况，只要以这个点为根进行树形依赖背包即可，具体做法不再赘述，通过二进制拆分可以做到 $O(nm\log d)$，通过单调队列可以做到 $O(nm)$。

## Solution

- 考虑某个点必选的情况，只要以这个点为根进行树形依赖背包即可，具体做法不再赘述，通过二进制拆分可以做到 $O(nm\log d)$，通过单调队列可以做到 $O(nm)$。
- 若某个点必然不选，那么去掉这个点也无妨，树将会分裂成若干独立的子树。

# Solution

- 考虑某个点必选的情况，只要以这个点为根进行树形依赖背包即可，具体做法不再赘述，通过二进制拆分可以做到 $O(nm\log d)$，通过单调队列可以做到 $O(nm)$。
- 若某个点必然不选，那么去掉这个点也无妨，树将会分裂成若干独立的子树。
- 这其实就是点分治的过程。

# Solution

- 考虑某个点必选的情况，只要以这个点为根进行树形依赖背包即可，具体做法不再赘述，通过二进制拆分可以做到 $O(nm\log d)$，通过单调队列可以做到 $O(nm)$。
- 若某个点必然不选，那么去掉这个点也无妨，树将会分裂成若干独立的子树。
- 这其实就是点分治的过程。
- 对这棵树进行点分治，重心要么选，要么不选，第一种情况用上述方法 DP 即可，第二种情况递归处理即可。

# Solution

- 考虑某个点必选的情况，只要以这个点为根进行树形依赖背包即可，具体做法不再赘述，通过二进制拆分可以做到 $O(nm\log d)$，通过单调队列可以做到 $O(nm)$。
- 若某个点必然不选，那么去掉这个点也无妨，树将会分裂成若干独立的子树。
- 这其实就是点分治的过程。
- 对这棵树进行点分治，重心要么选，要么不选，第一种情况用上述方法 DP 即可，第二种情况递归处理即可。
- 时间复杂度 $O(nm\log n)$。

# 课后习题

- Antonidas (HDU 5469)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)
- 烁烁的游戏 (BZOJ 4372)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)
- 烁烁的游戏 (BZOJ 4372)
- xmastree (BZOJ 2566)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)
- 烁烁的游戏 (BZOJ 4372)
- xmastree (BZOJ 2566)
- Styx (BZOJ 4623)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)
- 烁烁的游戏 (BZOJ 4372)
- xmastree (BZOJ 2566)
- Styx (BZOJ 4623)
- [Scoi2016] 幸运数字 (BZOJ 4568)

# 课后习题

- Antonidas (HDU 5469)
- [BeiJing2017] 树的难题 (BZOJ 4860)
- Query on the subtree (HDU 4918)
- 烁烁的游戏 (BZOJ 4372)
- xmastree (BZOJ 2566)
- Styx (BZOJ 4623)
- [Scoi2016] 幸运数字 (BZOJ 4568)
- More Health Points (ZOJ 3937)

# Thank you!