数据结构进阶

Claris

Hangzhou Dianzi University

2018 年 2 月 11 日

Overview

- 本节课继续介绍几种数据结构以及一些技巧：

Overview

- 本节课继续介绍几种数据结构以及一些技巧：
- 线段树。

Overview

- 本节课继续介绍几种数据结构以及一些技巧：
- 线段树。
- 倍增法。

Overview

- 本节课继续介绍几种数据结构以及一些技巧：
- 线段树。
- 倍增法。
- 树链剖分。

线段树

- 线段树是一种维护序列的二叉树。

线段树

- 线段树是一种维护序列的二叉树。
- 根节点表示区间 $[1, n]$。

线段树

- 线段树是一种维护序列的二叉树。
- 根节点表示区间 $[1, n]$。
- 对于任意一个表示区间 $[l, r]$ 的节点，若 $l < r$，则取 $mid = \lfloor \frac{l+r}{2} \rfloor$，该节点的左儿子为 $[l, mid]$，右儿子为 $[mid + 1, r]$。

线段树

- 线段树是一种维护序列的二叉树。
- 根节点表示区间 $[1, n]$。
- 对于任意一个表示区间 $[l, r]$ 的节点，若 $l < r$，则取 $mid = \lfloor \frac{l+r}{2} \rfloor$，该节点的左儿子为 $[l, mid]$，右儿子为 $[mid + 1, r]$。
- 若 $l = r$，则它为叶子。

线段树

- 线段树是一种维护序列的二叉树。
- 根节点表示区间 $[1, n]$。
- 对于任意一个表示区间 $[l, r]$ 的节点，若 $l < r$，则取 $mid = \lfloor \frac{l+r}{2} \rfloor$，该节点的左儿子为 $[l, mid]$，右儿子为 $[mid + 1, r]$。
- 若 $l = r$，则它为叶子。
- 树高 $O(\log n)$，节点数 $O(n)$。

线段树

- 指针表示法。

线段树

- 指针表示法。
- 每个点存储左右儿子的指针。

线段树

- 指针表示法。
- 每个点存储左右儿子的指针。
- 对于一棵树只需要记录根节点指针。

线段树

- 指针表示法。
- 每个点存储左右儿子的指针。
- 对于一棵树只需要记录根节点指针。
- 优点：可以推迟建树，减少空间复杂度。

线段树

- 堆式表示法。

线段树

- 堆式表示法。
- 根节点为 1，左儿子为 $x << 1$，右儿子为 $x << 1|1$，父亲为 $x >> 1$。

线段树

- 堆式表示法。
- 根节点为 1，左儿子为 $x << 1$，右儿子为 $x << 1|1$，父亲为 $x >> 1$。
- 优点：速度较快，不需要额外记录左右儿子。

线段树

- 每个点维护对应区间内的信息的和。

线段树

- 每个点维护对应区间内的信息的和。
- 信息需要满足可加性。

线段树

- 每个点维护对应区间内的信息的和。
- 信息需要满足可加性。
- 单点修改：从根开始，根据修改位置决定往左还是往右走，$O(\log n)$。

线段树

- 每个点维护对应区间内的信息的和。
- 信息需要满足可加性。
- 单点修改：从根开始，根据修改位置决定往左还是往右走，$O(\log n)$。
- 区间查询：从根开始遍历这棵树，当当前区间被完全包含时返回，$O(\log n)$。

线段树

- 区间更新：多维护懒惰标记。

线段树

- 区间更新：多维护懒惰标记。
- 区间更新时按照区间查询的方法将标记打到 $O(\log n)$ 个节点上。

线段树

- 区间更新：多维护懒惰标记。
- 区间更新时按照区间查询的方法将标记打到 $O(\log n)$ 个节点上。
- 每次经过一个点时，下放标记。

线段树

- 区间更新：多维护懒惰标记。
- 区间更新时按照区间查询的方法将标记打到 $O(\log n)$ 个节点上。
- 每次经过一个点时，下放标记。
- 注意打标记时如何使标记生效。

Quadruples of Points

平面上 n 个点集，每个集合恰好 4 个点。

Quadruples of Points

平面上 n 个点集，每个集合恰好 4 个点。

m 次询问，每次询问给出一个平行坐标轴的矩形，问是否对于每个集合，矩形都恰好框住了两个点。

Quadruples of Points

平面上 n 个点集，每个集合恰好 4 个点。

m 次询问，每次询问给出一个平行坐标轴的矩形，问是否对于每个集合，矩形都恰好框住了两个点。

- $n, m \leq 200000$。

Quadruples of Points

平面上 n 个点集，每个集合恰好 4 个点。

m 次询问，每次询问给出一个平行坐标轴的矩形，问是否对于每个集合，矩形都恰好框住了两个点。

- $n, m \leq 200000$。
- $|x|, |y| \leq 10^9$。

Quadruples of Points

平面上 n 个点集，每个集合恰好 4 个点。

m 次询问，每次询问给出一个平行坐标轴的矩形，问是否对于每个集合，矩形都恰好框住了两个点。

- $n, m \leq 200000$。
- $|x|, |y| \leq 10^9$。
- Source : XV Open Cup named after E.V. Pankratiev. GP of Three Capitals

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。
- 给每个集合一个随机的权值 v，该集合中任意一个点的权值都为 v。

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。
- 给每个集合一个随机的权值 v，该集合中任意一个点的权值都为 v。
- 若询问矩形内部所有点的权值之和恰好等于 $2\sum v$，则可行。

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。
- 给每个集合一个随机的权值 v，该集合中任意一个点的权值都为 v。
- 若询问矩形内部所有点的权值之和恰好等于 $2\sum v$，则可行。
- 将矩形拆成两个询问，去掉左边界的限制。

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。
- 给每个集合一个随机的权值 v，该集合中任意一个点的权值都为 v。
- 若询问矩形内部所有点的权值之和恰好等于 $2\sum v$，则可行。
- 将矩形拆成两个询问，去掉左边界的限制。
- 从左往右依次加入每个点，线段树维护纵方向区间和。

Solution

- 坐标可以离散化，使得坐标不超过 $O(n+m)$。
- 给每个集合一个随机的权值 v，该集合中任意一个点的权值都为 v。
- 若询问矩形内部所有点的权值之和恰好等于 $2\sum v$，则可行。
- 将矩形拆成两个询问，去掉左边界的限制。
- 从左往右依次加入每个点，线段树维护纵方向区间和。
- 时间复杂度 $O(n \log n)$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, \ldots, a_n$，支持三种操作：

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, \ldots, a_n$，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, \ldots, a_r$ 全部乘以 v。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

3 l r：输出 $(a_l \times a_{l+1} \times ... \times a_r) \bmod 1,000,000,007$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

3 l r：输出 $(a_l \times a_{l+1} \times ... \times a_r) \bmod 1,000,000,007$。

- $n, m \leq 100000$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

3 l r：输出 $(a_l \times a_{l+1} \times ... \times a_r) \bmod 1,000,000,007$。

- $n, m \leq 100000$。
- $1 \leq a_i, v \leq 1000$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

3 l r：输出 $(a_l \times a_{l+1} \times ... \times a_r) \bmod 1,000,000,007$。

- $n, m \leq 100000$。
- $1 \leq a_i, v \leq 1000$。
- $1 \leq k \leq 10^9$。

Yet Another Data Structure Problem

维护正整数序列 $a_1, a_2, ..., a_n$ ，支持三种操作：

1 l r v：将 $a_l, a_{l+1}, ..., a_r$ 全部乘以 v。

2 l r k：将 $a_l, a_{l+1}, ..., a_r$ 修改为 $a_l^k, a_{l+1}^k, ..., a_r^k$。

3 l r：输出 $(a_l \times a_{l+1} \times ... \times a_r) \bmod 1,000,000,007$。

- $n, m \leq 100000$。
- $1 \leq a_i, v \leq 1000$。
- $1 \leq k \leq 10^9$。
- Source： ZOJ 3998

Permutation

给定一个长度为 n 的排列 $a_1, a_2, \ldots, a_n$。

Permutation

给定一个长度为 n 的排列 $a_1, a_2, \ldots, a_n$。

请统计有多少个区间 $[l, r]$ 满足 $1 \leq l \leq r \leq n$，且将 $a_l, a_{l+1}, \ldots, a_r$ 排序后是一个公差为 1 的等差数列。

Permutation

给定一个长度为 n 的排列 $a_1, a_2, \ldots, a_n$。

请统计有多少个区间 $[l, r]$ 满足 $1 \leq l \leq r \leq n$，且将 $a_l, a_{l+1}, \ldots, a_r$ 排序后是一个公差为 1 的等差数列。

- $n \leq 100000$。

倍增法

- 一般是预处理出 $2^0, 2^1, 2^2, \ldots, 2^{\log n}$ 的信息。

倍增法

- 一般是预处理出 $2^0, 2^1, 2^2, \ldots, 2^{\log n}$ 的信息。
- 将查询分解成 $O(\log n)$ 个 2^k 的信息的和。

倍增法

- 一般是预处理出 $2^0, 2^1, 2^2, \ldots, 2^{\log n}$ 的信息。
- 将查询分解成 $O(\log n)$ 个 2^k 的信息的和。
- 即将 n 二进制中 1 的部分累加。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i + 2^j - 1]$ 的最值。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i + 2^j - 1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i+2^j-1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。
- 递推式：$f_{i,j} = \max(f_{i,j-1}, f_{i+2^{j-1},j-1})$。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i+2^j-1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。
- 递推式：$f_{i,j} = \max(f_{i,j-1}, f_{i+2^{j-1},j-1})$。
- 预处理 $O(n \log n)$。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i+2^j-1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。
- 递推式：$f_{i,j} = \max(f_{i,j-1}, f_{i+2^{j-1},j-1})$。
- 预处理 $O(n \log n)$。
- 对于询问 $[l, r]$，设 $k = \lfloor \log(r-l+1) \rfloor$，则 $ans = \max(f_{l,k}, f_{r-2^k+1,k})$。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i+2^j-1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。
- 递推式：$f_{i,j} = \max(f_{i,j-1}, f_{i+2^{j-1},j-1})$。
- 预处理 $O(n \log n)$。
- 对于询问 $[l, r]$，设 $k = \lfloor \log(r-l+1) \rfloor$，则 $ans = \max(f_{l,k}, f_{r-2^k+1,k})$。
- 这是因为最值允许重复统计。

ST 表

- 对于没有修改操作的区间最值的询问，可以使用 ST 表。
- ST 表基于倍增法，设 $f_{i,j}$ 表示 $[i, i+2^j-1]$ 的最值。
- 边界值：$f_{i,0} = a_i$。
- 递推式：$f_{i,j} = \max(f_{i,j-1}, f_{i+2^{j-1},j-1})$。
- 预处理 $O(n \log n)$。
- 对于询问 $[l, r]$，设 $k = \lfloor \log(r-l+1) \rfloor$，则 $ans = \max(f_{l,k}, f_{r-2^k+1,k})$。
- 这是因为最值允许重复统计。
- 询问复杂度 $O(1)$。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。
- 递推式：$f_{i,j} = f_{f_{i,j-1},j-1}$。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。
- 递推式：$f_{i,j} = f_{f_{i,j-1},j-1}$。
- 预处理 $O(n \log n)$。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。
- 递推式：$f_{i,j} = f_{f_{i,j-1},j-1}$。
- 预处理 $O(n \log n)$。
- 对于询问 x, y，首先将 x 和 y 调到同一深度。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。
- 递推式：$f_{i,j} = f_{f_{i,j-1},j-1}$。
- 预处理 $O(n \log n)$。
- 对于询问 x, y，首先将 x 和 y 调到同一深度。
- 再同时向上倍增枚举祖先，若不相同则往上走，最后一定只差一步就到 LCA。

最近公共祖先

- 最近公共祖先 (LCA) 即有根树上两点路径上深度最小的点。
- 倍增法求 LCA，设 $f_{i,j}$ 表示 i 往上走 2^j 步到达的点。
- 边界值：$f_{i,0} = father_i$。
- 递推式：$f_{i,j} = f_{f_{i,j-1},j-1}$。
- 预处理 $O(n \log n)$。
- 对于询问 x, y，首先将 x 和 y 调到同一深度。
- 再同时向上倍增枚举祖先，若不相同则往上走，最后一定只差一步就到 LCA。
- 询问复杂度 $O(\log n)$。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。
- 求出 LCA 后可以转化为两个询问，且保证 y 是 x 的祖先。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。
- 求出 LCA 后可以转化为两个询问，且保证 y 是 x 的祖先。
- $g_{i,j}$ 表示 i 往上 2^j 条边的最大值。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。
- 求出 LCA 后可以转化为两个询问，且保证 y 是 x 的祖先。
- $g_{i,j}$ 表示 i 往上 2^j 条边的最大值。
- $g_{i,j} = \max(g_{i,j-1}, g_{f_{i,j-1},j-1})$。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。
- 求出 LCA 后可以转化为两个询问，且保证 y 是 x 的祖先。
- $g_{i,j}$ 表示 i 往上 2^j 条边的最大值。
- $g_{i,j} = \max(g_{i,j-1}, g_{f_{i,j-1},j-1})$。
- 通过倍增法分解成 $O(\log n)$ 个 g 求最大值。

树上路径统计

- 求树上两点间 x, y 路径上边权最大值。
- 求出 LCA 后可以转化为两个询问，且保证 y 是 x 的祖先。
- $g_{i,j}$ 表示 i 往上 2^j 条边的最大值。
- $g_{i,j} = \max(g_{i,j-1}, g_{f_{i,j-1},j-1})$。
- 通过倍增法分解成 $O(\log n)$ 个 g 求最大值。
- 询问复杂度 $O(\log n)$。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。
- 树链剖分是将树剖分成一条条不相交的链。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。
- 树链剖分是将树剖分成一条条不相交的链。
- 设 $size_x$ 表示 x 子树内节点数，每个点选择 $size$ 最大的儿子连边，称为重儿子。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。
- 树链剖分是将树剖分成一条条不相交的链。
- 设 $size_x$ 表示 x 子树内节点数，每个点选择 $size$ 最大的儿子连边，称为重儿子。
- 将与重儿子连的边称为重边，其余称为轻边。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。
- 树链剖分是将树剖分成一条条不相交的链。
- 设 $size_x$ 表示 x 子树内节点数，每个点选择 $size$ 最大的儿子连边，称为重儿子。
- 将与重儿子连的边称为重边，其余称为轻边。
- 一个点到根最多经过 $O(\log n)$ 条轻边。

树链剖分

- 倍增法求 LCA 的弊端是空间大、速度慢。
- 树链剖分是将树剖分成一条条不相交的链。
- 设 $size_x$ 表示 x 子树内节点数，每个点选择 $size$ 最大的儿子连边，称为重儿子。
- 将与重儿子连的边称为重边，其余称为轻边。
- 一个点到根最多经过 $O(\log n)$ 条轻边。
- 证明：每经过一条轻边 $size$ 至少除以 2，否则不可能不是重儿子。

树链剖分

- 树链剖分求 LCA。

树链剖分

- 树链剖分求 LCA。
- 求出每个点 x 所在重链顶端的点 top_x。

树链剖分

- 树链剖分求 LCA。
- 求出每个点 x 所在重链顶端的点 top_x。
- 对于求 x, y 的 LCA，若 x 和 y 在同一条重链中，则结果显然。

树链剖分

- 树链剖分求 LCA。
- 求出每个点 x 所在重链顶端的点 top_x。
- 对于求 x, y 的 LCA，若 x 和 y 在同一条重链中，则结果显然。
- 否则将 top 深度较大的那一个点往上跳到 top。

树链剖分

- 树链剖分求 LCA。
- 求出每个点 x 所在重链顶端的点 top_x。
- 对于求 x, y 的 LCA，若 x 和 y 在同一条重链中，则结果显然。
- 否则将 top 深度较大的那一个点往上跳到 top。
- 时间复杂度 $O(\log n)$，常数很小。

树链剖分

- 树链剖分求 LCA。
- 求出每个点 x 所在重链顶端的点 top_x。
- 对于求 x, y 的 LCA，若 x 和 y 在同一条重链中，则结果显然。
- 否则将 top 深度较大的那一个点往上跳到 top。
- 时间复杂度 $O(\log n)$，常数很小。
- 同时树链剖分将一棵树分解成了一条条链，可以很方便地将路径维护问题转化为序列维护问题。

Carpet

n 个点的无根树，要求将其画在 $10^6 \times 20$ 的网格中。

Carpet

n 个点的无根树，要求将其画在 $10^6 \times 20$ 的网格中。不允许一个格子放多个点。

Carpet

n 个点的无根树，要求将其画在 $10^6 \times 20$ 的网格中。

不允许一个格子放多个点。

不允许树边相交。

Carpet

n 个点的无根树，要求将其画在 $10^6 \times 20$ 的网格中。

不允许一个格子放多个点。

不允许树边相交。

- $n \leq 100000$。

Carpet

n 个点的无根树，要求将其画在 $10^6 \times 20$ 的网格中。

不允许一个格子放多个点。

不允许树边相交。

- $n \leq 100000$。
- Source：NEERC 2017 Moscow Subregional Contest

Solution

- 树链剖分。

Solution

- 树链剖分。
- 重边往下挂。

Solution

- 树链剖分。
- 重边往下挂。
- 轻边往右挂。

Solution

- 树链剖分。
- 重边往下挂。
- 轻边往右挂。
- 时间复杂度 $O(n)$。

遥远的国度

给定一棵以 1 为根的树，每个点有权值 v_i，支持两种操作：

遥远的国度

给定一棵以 1 为根的树，每个点有权值 v_i，支持两种操作：

1 x y k：将 x 到 y 路径上所有点权值都修改为 k。

遥远的国度

给定一棵以 1 为根的树，每个点有权值 v_i，支持两种操作：

1 x y k：将 x 到 y 路径上所有点权值都修改为 k。

2 x：查询 x 子树中点权的最小值。

遥远的国度

给定一棵以 1 为根的树，每个点有权值 v_i，支持两种操作：

1 x y k：将 x 到 y 路径上所有点权值都修改为 k。

2 x：查询 x 子树中点权的最小值。

- $n, m \leq 100000$。

遥远的国度

给定一棵以 1 为根的树，每个点有权值 v_i，支持两种操作：

1 x y k：将 x 到 y 路径上所有点权值都修改为 k。

2 x：查询 x 子树中点权的最小值。

- $n, m \leq 100000$。
- Source： BZOJ 3083

Laminar Family

给定一棵 n 个点的树，以及 m 个树上的点集。

Laminar Family

给定一棵 n 个点的树，以及 m 个树上的点集。

每个集合用 a_i 和 b_i 描述，表示 a_i 到 b_i 路径上的所有点。

Laminar Family

给定一棵 n 个点的树，以及 m 个树上的点集。

每个集合用 a_i 和 b_i 描述，表示 a_i 到 b_i 路径上的所有点。

你需要判断，是否对于任意两个集合，要么它们没有交集，要么它们是包含关系。

Laminar Family

给定一棵 n 个点的树，以及 m 个树上的点集。

每个集合用 a_i 和 b_i 描述，表示 a_i 到 b_i 路径上的所有点。

你需要判断，是否对于任意两个集合，要么它们没有交集，要么它们是包含关系。

- $n, m \leq 100000$。

Laminar Family

给定一棵 n 个点的树，以及 m 个树上的点集。

每个集合用 a_i 和 b_i 描述，表示 a_i 到 b_i 路径上的所有点。

你需要判断，是否对于任意两个集合，要么它们没有交集，要么它们是包含关系。

- $n, m \leq 100000$。
- Source： NEERC 2017

Phone Call

给定一棵 n 个点的树以及 m 条电话线路。

Phone Call

给定一棵 n 个点的树以及 m 条电话线路。

每条电话线路意为两条树链上的点两两之间可以通过 w 的代价进行通话。

Phone Call

给定一棵 n 个点的树以及 m 条电话线路。

每条电话线路意为两条树链上的点两两之间可以通过 w 的代价进行通话。

求 1 号点能直接或间接通话到的人数的最大值以及在此基础上的最小代价。

Phone Call

给定一棵 n 个点的树以及 m 条电话线路。

每条电话线路意为两条树链上的点两两之间可以通过 w 的代价进行通话。

求 1 号点能直接或间接通话到的人数的最大值以及在此基础上的最小代价。

- $n, m \leq 100000$。

Phone Call

给定一棵 n 个点的树以及 m 条电话线路。

每条电话线路意为两条树链上的点两两之间可以通过 w 的代价进行通话。

求 1 号点能直接或间接通话到的人数的最大值以及在此基础上的最小代价。

- $n, m \leq 100000$。
- Source : 2017 Multi-University Training Contest 4

题目提交

课上例题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33145

课后习题：

http://acm.hdu.edu.cn/diy/contest_show.php?cid=33146

密码：

G*&GSF&*t387tr

Thank you!