数据结构选讲

QQH

January 28, 2023

1 前言

2 ST 表

3 树状数组

4 线段树
- 可持久化线段树
- 线段树合并
- 区间最值问题
- 李超线段树

5 杂题

Will liuhengxi achieve LGM in grade 9?

By dXqwq, history, 3 weeks ago,

> As far as I know, he is in grade 9 now, which is one year younger than orzdevinwang.
>
> He is gaining rating in recent contests and previous LGMs in grade 9 are djq-cpp and orzdevinwang(comment below if you know more).
>
> Meanwhile I'm stuck between 2600~2700, hope I will become stronger next year o(>_<)o

+49 dXqwq 3 weeks ago 5

Comments (5)

Write comment?

N_z__ — 3 weeks ago, # | +34

As a classmate of liuhengxi,I think he will.

→ Reply

moonrise__ — 3 weeks ago, # ^ | +68

As a classmate of liuhengxi, I think he wants a girlfriend.

→ Reply

User_Carrot — 3 weeks ago, # ^ | +20

As a classmate and also a fan of liuhengxi, I think he will. And he wants a girlfriend btw.

→ Reply

Codeforces account: LIUHENGXI INTERNATIONAL GRANDMASTER 2891

找 npy，要求：

令 c_j 为 CSP-J 中获得过的最高分，c_s 为 CSP-S 中获得过的最高分，r_c 为 Codeforces rating，r_a 为 Atcoder rating，b 为出生年份（x 年的第 y 天为 $x+\dfrac{y}{365}$）。

$$s_{oi} = \ln\left(\exp\left(\frac{c_j-500}{80}\right)+\exp\left(\frac{c_s-260}{50}\right)+\exp\left(\frac{r_c-2000}{400}\right)+\exp\left(\frac{r_a-1500}{600}\right)\right).$$

$$d=\frac{6}{11}(b-2007)-1.$$

$$s_{birth}=(\cosh(d+2\arctan d))^{-1}.$$

$$s=10\cdot\max\{1+\min\{s_{oi},3\},0\}+40\cdot s_{birth}+20\geq 70$$

为了公式美观，记 $\exp x = e^x$.

ST 表相信大家都会，主要解决 RMQ 问题。

ST 表相信大家都会，主要解决 RMQ 问题。
下面主要看几道题。

CF713D Animals and Puzzle

给定一个 $n \times m$ 的 0/1 矩阵。t 次询问，每次询问左上角为 (x_1, y_1)，右下角为 (x_2, y_2) 的子矩阵中, 全为 1 的最长正方形边长。

$n, m \leq 1000, q \leq 10^6$

CF713D Animals and Puzzle

设 $f_{i,j}$ 为以 (i, j) 为左上角的全为 1 的最长正方形边长。

CF713D Animals and Puzzle

设 $f_{i,j}$ 为以 (i,j) 为左上角的全为 1 的最长正方形边长。
二分答案，问题变成询问子矩阵中 $f_{i,j}$ 的最大值。

CF713D Animals and Puzzle

设 $f_{i,j}$ 为以 (i,j) 为左上角的全为 1 的最长正方形边长。
二分答案，问题变成询问子矩阵中 $f_{i,j}$ 的最大值。
二维 ST 表维护。

CF713D Animals and Puzzle

设 $f_{i,j}$ 为以 (i,j) 为左上角的全为 1 的最长正方形边长。
二分答案，问题变成询问子矩阵中 $f_{i,j}$ 的最大值。
二维 ST 表维护。
时间复杂度：$O(nm\log n\log m + T\log\min(n,m))$

soj 1589 随机

有一个长为 n，值域在 $[1, V]$ 的正整数序列 a，有 m 次修改，每次将 a_x 赋值为 v。

定义 $f(l, r) = (\max\{a_l, a_{l+1}, \dots, a_r\} - \min\{a_l, a_{l+1}, \dots, a_r\})^2$，你需要在每次修改后求出 $\sum_{l=1}^{n} \sum_{r=l}^{n} [r - l + 1 = 2^k] f(l, r)$。

$1 \le n, m \le 5 \times 10^5, 1 \le V \le 2 \times 10^5$，保证 a_i, v, x 独立均匀随机。

soj 1589 随机

只要你敢打暴力，你就过了！

soj 1589 随机

只要你敢打暴力，你就过了！

先用 st 表维护区间最小值和最大值，考虑一次赋值操作，分别把最大值和最小值被它影响到的区间找出来，然后暴力修改在这个区间里的对应 st 表，顺便更新答案。

soj 1589 随机

只要你敢打暴力，你就过了！

先用 st 表维护区间最小值和最大值，考虑一次赋值操作，分别把最大值和最小值被它影响到的区间找出来，然后暴力修改在这个区间里的对应 st 表，顺便更新答案。

因为数据保证随机，所以用笛卡尔树分析可知这种区间长度为 $O(\log n)$ 级别的。时间复杂度 $O((n+q)\log n)$。

ST 表可以解决可重复贡献问题，所以除了 RMQ，例如区间按位与，区间 gcd 的问题，ST 表都可以解决。

树状数组相信大家都会。

树状数组相信大家都会。

```
struct TREE{
    int tr[N];
    void add(int x,int v){for(;x<=n;x+=(x&-x))tr[x]+=v;}
    int ask(int x){int res=0;for(;x>=1;x-=(x&-x))res+=tr[x];return res;}
}TR;
```

loj 131: 区间修改，单点查询

给定数列 $a_{1\dots n}$， q 个操作，操作有两类：
1 l r x：将 $[l, r]$ 内的数加 x。
2 x：查询 a_x。
$1 \le n, q \le 10^6$

loj 132: 区间修改，区间查询

给定数列 $a_{1\dots n}$，q 个操作，操作有两类：
1 l r x：将 $[l, r]$ 内的数加 x。
2 l r：查询 $[l, r]$ 区间内 a_i 的和。
$1 \le n, q \le 10^6$

loj 132: 区间修改，区间查询

维护 a 的差分数组 b，考虑对 a 的前缀求和。

$$\begin{aligned}\sum_{i=1}^{x} a_i &= \sum_{i=1}^{x}\sum_{j=1}^{i} b_j \\ &= \sum_{i=1}^{x} b_i \times (x-i+1) \\ &= \sum_{i=1}^{x} b_i \times (x+1) - \sum_{i=1}^{x} b_i \times i\end{aligned}$$

用 2 棵树状数组维护即可。

查询第 k 小

给定 q 个操作，和初始为空的可重集合 S，操作有两类：

1. 给定 x，在 S 中插入 x。
2. 给定 k，查询 S 中的第 k 小元素。

$1 \leq x, q \leq 10^6$

查询第 k 小

找第 k 小即为找最小的 x，满足 $\sum_{i=1}^{x} a_i \geq k$。

查询第 k 小

找第 k 小即为找最小的 x，满足 $\sum_{i=1}^{x} a_i \geq k$。

考虑到树状数组是按照 2 的幂次划分的，所以每次扩大 2 的幂次长度来找 x。流程如下：

查询第 k 小

找第 k 小即为找最小的 x，满足 $\sum_{i=1}^{x} a_i \geq k$。

考虑到树状数组是按照 2 的幂次划分的，所以每次扩大 2 的幂次长度来找 x。流程如下：

1. $dep = \log_2 n$。
2. 计算 $o = \sum_{i=x+1}^{x+2^dep} a_i$。
3. 如果 $now + o < k$，x 加上 2^dep。
4. 将 dep 减 1，回到步骤二，直到 dep 为 0。

查询第 k 小

找第 k 小即为找最小的 x，满足 $\sum_{i=1}^{x} a_i \geq k$。

考虑到树状数组是按照 2 的幂次划分的，所以每次扩大 2 的幂次长度来找 x。流程如下：

1. $dep = \log_2 n$。
2. 计算 $o = \sum_{i=x+1}^{x+2^dep} a_i$。
3. 如果 $now + o < k$，x 加上 2^dep。
4. 将 dep 减 1，回到步骤二，直到 dep 为 0。

所以权值树状数组可以解决平衡树的部分操作。

二维树状数组

给定一个矩阵 a，q 次操作，操作有两类：
1. 将 $a_{x,y}$ 加 z。
2. 查询 $\sum_{1\le i\le x,1\le j\le y} a_{i,j}$。
$1 \le x, y \le 5000, 1 \le q \le 10^5$

二维树状数组

给定一个矩阵 a，q 次操作，操作有两类：

1. 将 $a_{x,y}$ 加 z。
2. 查询 $\sum_{1\le i\le x,1\le j\le y} a_{i,j}$。

$1 \le x, y \le 5000, 1 \le q \le 10^5$

同理，二维树状数组也可以类似一维树状数组的推导出区间修改，区间查询。大家可以自行尝试。

线段树相信大家都会。

线段树相信大家都会。下面直接看几道题。

luogu P3373 线段树 2

如题，已知一个数列，你需要进行下面三种操作：
1. 将某区间每一个数乘上 x
2. 将某区间每一个数加上 x
3. 求出某区间每一个数的和
$1 \leq n, m \leq 10^5$

luogu P3373 线段树 2

如题，已知一个数列，你需要进行下面三种操作：
1. 将某区间每一个数乘上 x
2. 将某区间每一个数加上 x
3. 求出某区间每一个数的和

$1 \leq n, m \leq 10^5$

线段树，分别维护乘法和加法的 $lazytag$，先乘再加。

普通平衡树

您需要写一种数据结构，来维护一些数，其中需要提供以下操作：
1. 插入 x 数
2. 删除 x 数 (若有多个相同的数，因只删除一个)
3. 查询 x 数的排名 (排名定义为比当前数小的数的个数 $+1$)
4. 查询排名为 x 的数
5. 求 x 的前驱 (前驱定义为小于 x，且最大的数)
6. 求 x 的后继 (后继定义为大于 x，且最小的数)

$1 \leq n \leq 10^5, 1 \leq x \leq 10^9$

普通平衡树

您需要写一种数据结构，来维护一些数，其中需要提供以下操作：
1. 插入 x 数
2. 删除 x 数 (若有多个相同的数，因只删除一个)
3. 查询 x 数的排名 (排名定义为比当前数小的数的个数 $+1$)
4. 查询排名为 x 的数
5. 求 x 的前驱 (前驱定义为小于 x，且最大的数)
6. 求 x 的后继 (后继定义为大于 x，且最小的数)

$1 \leq n \leq 10^5, 1 \leq x \leq 10^9$

对元素离散化，用权值线段树维护，可在线段树上二分解决。

普通平衡树 2

您需要写一种数据结构，来维护一些数，其中需要提供以下操作：
1. 插入 x 数
2. 删除 x 数 (若有多个相同的数，因只删除一个)
3. 查询 x 数的排名 (排名定义为比当前数小的数的个数 $+1$)
4. 查询排名为 x 的数
5. 求 x 的前驱 (前驱定义为小于 x，且最大的数)
6. 求 x 的后继 (后继定义为大于 x，且最小的数)

强制在线。

$1 \leq n \leq 10^5, 1 \leq x \leq 10^9$

普通平衡树 2

您需要写一种数据结构，来维护一些数，其中需要提供以下操作：

1. 插入 x 数
2. 删除 x 数 (若有多个相同的数，因只删除一个)
3. 查询 x 数的排名 (排名定义为比当前数小的数的个数 $+1$)
4. 查询排名为 x 的数
5. 求 x 的前驱 (前驱定义为小于 x，且最大的数)
6. 求 x 的后继 (后继定义为大于 x，且最小的数)

强制在线。

$1 \leq n \leq 10^5, 1 \leq x \leq 10^9$

普通平衡树 2

因为强制在线，所以没法离散化元素。考虑动态开点线段树，只创建需要访问的节点。时空复杂度为 $O(n\log x)$。

普通平衡树 2

因为强制在线，所以没法离散化元素。考虑动态开点线段树，只创建需要访问的节点。时空复杂度为 $O(n \log x)$。

在需要访问的节点远小于整棵树大小，或需要很多棵树的时候，就需要动态开点。

1 前言

2 ST 表

3 树状数组

4 线段树
- 可持久化线段树
- 线段树合并
- 区间最值问题
- 李超线段树

5 杂题

可持久化线段树

维护一个序列 a，支持三种操作：

1. 区间加。
2. 回到第 t 次修改后。
3. 查询第 t 次修改后的区间和。

$1 \leq n, m \leq 10^5$

可持久化线段树

维护一个序列 a，支持三种操作：
1. 区间加。
2. 回到第 t 次修改后。
3. 查询第 t 次修改后的区间和。

$1 \leq n, m \leq 10^5$

对线段树可持久化，每次把修改的链拉出来重新建立节点，保留原来的信息并修改。

区间第 k 小

给定序列 a，每次查询区间 $[l, r]$ 的第 k 小值。

$1 \le n, m \le 10^5$

区间第 k 小

给定序列 a，每次查询区间 $[l,r]$ 的第 k 小值。

$1 \le n,m \le 10^5$

离散化，对权值线段树可持久化，然后再线段树上二分，维护前缀和就可以算出区间 $[l,r]$ 中的出现次数了。

CF484E Sign on Fence

给定一个长度为 n 的数列，有 m 次询问，询问形如 $l\ r\ k$。
要你在区间 $[l, r]$ 内选一个长度为 k 的区间，求区间最小数的最大值。
$1 \le n, m \le 10^5$

CF484E Sign on Fence

离散化，考虑二分答案 mid，我们令 $b_i = [a_i >= mid]$，发现只要区间 $[l, r]$ 中最长连续 1 段长度 $\geq k$ 就是可行的。

CF484E Sign on Fence

离散化，考虑二分答案 mid，我们令 $b_i = [a_i >= mid]$，发现只要区间 $[l, r]$ 中最长连续 1 段长度 $\geq k$ 就是可行的。

用可持久化线段树维护每一个 mid 时的情况即可。时间复杂度 $O(m \log^2 n)$。

CF484E Sign on Fence

离散化，考虑二分答案 mid，我们令 $b_i = [a_i >= mid]$，发现只要区间 $[l, r]$ 中最长连续 1 段长度 $\geq k$ 就是可行的。

用可持久化线段树维护每一个 mid 时的情况即可。时间复杂度 $O(m \log^2 n)$。

好像有 $1 \log$ 做法，但这个做法比较好想。

1 前言

2 ST 表

3 树状数组

4 线段树
- 可持久化线段树
- 线段树合并
- 区间最值问题
- 李超线段树

5 杂题

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：
1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：
1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。
2. b 有 p 位置，而 a 没有。那么新的线段树 p 位置赋成 b，返回。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：
1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。
2. b 有 p 位置，而 a 没有。那么新的线段树 p 位置赋成 b，返回。
3. 如果 p 是叶子结点，将两个叶子节点的信息合并，返回。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：
1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。
2. b 有 p 位置，而 a 没有。那么新的线段树 p 位置赋成 b，返回。
3. 如果 p 是叶子结点，将两个叶子节点的信息合并，返回。
4. 递归处理左子树和右子树。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：

1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。
2. b 有 p 位置，而 a 没有。那么新的线段树 p 位置赋成 b，返回。
3. 如果 p 是叶子结点，将两个叶子节点的信息合并，返回。
4. 递归处理左子树和右子树。
5. 用左右子树信息更新当前节点。

线段树合并

线段树合并，就是把 2 棵线段树的信息合并。
其实思想很简单，假设考虑到线段树 a, b 的 p 位置：
1. a 有 p 位置，而 b 没有。那么新的线段树 p 位置赋成 a，返回。
2. b 有 p 位置，而 a 没有。那么新的线段树 p 位置赋成 b，返回。
3. 如果 p 是叶子结点，将两个叶子节点的信息合并，返回。
4. 递归处理左子树和右子树。
5. 用左右子树信息更新当前节点。
6. 将新的线段树 p 位置赋成 a。

线段树合并

```cpp
int merge(int p,int q,int l,int r){
    if(!p||!q)return p^q;
    if(l==r){
        //...
        return p;
    }
    int mid=(l+r)>>1;
    tr[p].ls=merge(tr[p].ls,tr[q].ls,l,mid);
    tr[p].rs=merge(tr[p].rs,tr[q].rs,mid+1,r);
    push_up(p);
    return p;
}
```

luogu P4556

有一颗 n 个节点的树，每个节点有一个可重集合。m 次操作，每次给出 x, y, z，将 z 加入 x 到 y 路径上所有节点的集合中。

最后对于 $i = 1, 2, \ldots, n$，输出 i 号节点的可重集合中出现次数最多的数。集合为空就输出 0。

$1 \leq n, m, z \leq 10^5$

luogu P4556

对每一个节点维护一课权值线段树，记录每一个数的出现次数。

luogu P4556

对每一个节点维护一课权值线段树，记录每一个数的出现次数。然后把每一次操作树上差分，直接向上合并线段树即可。

1 前言

2 ST 表

3 树状数组

4 线段树
- 可持久化线段树
- 线段树合并
- 区间最值问题
- 李超线段树

5 杂题

区间最值问题是吉老师在 2016 年国家集训队论文中系统提出的，所以大家也可以去看论文来学习。

HDU5306 Gorgeous Sequence

维护一个序列 a:

1. 区间与 x 取 $\min$
2. 查询区间最大值
3. 查询区间和

$1 \leq n, m \leq 10^5$

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。
但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。
但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。
所以再记录区间最大值的个数 cnt，次大值 $Max2$ 和区间和 sum。
接下来考虑与 x 取 min 的操作：

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。
但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。
所以再记录区间最大值的个数 cnt，次大值 $Max2$ 和区间和 sum。
接下来考虑与 x 取 min 的操作：

1. 如果 $Max \leq x$，不用操作。

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。
但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。
所以再记录区间最大值的个数 cnt，次大值 $Max2$ 和区间和 sum。
接下来考虑与 x 取 min 的操作：
1. 如果 $Max \leq x$，不用操作。
2. 如果 $Max2 \leq x \leq Max$，发现只对区间内的最大值有影响，所以 sum 加上 $cnt(x - Max)$，更新 Max 为 x，打标记。

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。

但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。

所以再记录区间最大值的个数 cnt，次大值 $Max2$ 和区间和 sum。

接下来考虑与 x 取 min 的操作：

1. 如果 $Max \leq x$，不用操作。

2. 如果 $Max2 \leq x \leq Max$，发现只对区间内的最大值有影响，所以 sum 加上 $cnt(x - Max)$，更新 Max 为 x，打标记。

3. 如果 $x \leq Max2$，向下递归更新。

HDU5306 Gorgeous Sequence

因为有取 min 操作，所以考虑线段树记录区间最大值 Max。

但是发现比如只在 $Max \geq x$ 时更新的复杂度可以被卡到 $O(n^2)$。

所以再记录区间最大值的个数 cnt，次大值 $Max2$ 和区间和 sum。

接下来考虑与 x 取 min 的操作：

1. 如果 $Max \leq x$，不用操作。

2. 如果 $Max2 \leq x \leq Max$，发现只对区间内的最大值有影响，所以 sum 加上 $cnt(x - Max)$，更新 Max 为 x，打标记。

3. 如果 $x \leq Max2$，向下递归更新。

可以证明该做法的时间复杂度是 $O(m \log n)$ 的，具体证明还是看论文吧。

题

维护一个序列 a:

1. 区间与 x 取 $\min$
2. 区间加 x（x 可为负数）
3. 查询区间和

$1 \le n, m \le 10^5$

题

这道题沿用上一道题的思路，区间加减再记录一个 tag 即可。

题

这道题沿用上一道题的思路，区间加减再记录一个 tag 即可。
论文证明了时间复杂度是 $O(m\log^2 n)$。

1 前言

2 ST 表

3 树状数组

4 线段树
- 可持久化线段树
- 线段树合并
- 区间最值问题
- 李超线段树

5 杂题

李超线段树

李超线段树是一种特殊的线段树。下面先考虑一个问题。

维护一个二维平面直角坐标系，支持两种操作：
1. 加入一条线段
2. 查询平面上与直线 $x = x_0$ 相交的线段中交点 y 的最大或最小值。

因为有区间修改，所以考虑还是在每一个节点记录懒标记（这里懒标记为一条线段）。

因为有区间修改，所以考虑还是在每一个节点记录懒标记（这里懒标记为一条线段）。

现在考虑加入一条线段 u，如果当前区间没有标记，那直接将标记赋为 u。

因为有区间修改，所以考虑还是在每一个节点记录懒标记（这里懒标记为一条线段）。

现在考虑加入一条线段 u，如果当前区间没有标记，那直接将标记赋为 u。

否则，设原标记为 v。由于标记无法合并，所以考虑递归下传标记。

因为有区间修改，所以考虑还是在每一个节点记录懒标记（这里懒标记为一条线段）。

现在考虑加入一条线段 u，如果当前区间没有标记，那直接将标记赋为 u。

否则，设原标记为 v。由于标记无法合并，所以考虑递归下传标记。

按照是否能被 u 更新将区间划分成两个子区间，其中肯定有一个被左或右区间包含。即两条线段中有一条线段只会成为左区间或右区间的答案，那么我们递归更新对应儿子，另一条线段作为整个区间的懒标记。

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

1. 如果在左端点处 u 更优，那 u 只能更新左区间，递归更新左区间。

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

1. 如果在左端点处 u 更优，那 u 只能更新左区间，递归更新左区间。

2. 如果在右端点处 u 更优，那 u 只能更新右区间，递归更新右区间。

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

1. 如果在左端点处 u 更优，那 u 只能更新左区间，递归更新左区间。
2. 如果在右端点处 u 更优，那 u 只能更新右区间，递归更新右区间。
3. 否则，u 无法更新区间。

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

1. 如果在左端点处 u 更优，那 u 只能更新左区间，递归更新左区间。
2. 如果在右端点处 u 更优，那 u 只能更新右区间，递归更新右区间。
3. 否则，u 无法更新区间。

最后懒标记设为 v。

（u 和 v 在中点相等的情况也可以同样处理）

不妨假设 v 在中点比 u 要优（如果不是直接交换 u 和 v 即可），具体步骤如下：

1. 如果在左端点处 u 更优，那 u 只能更新左区间，递归更新左区间。

2. 如果在右端点处 u 更优，那 u 只能更新右区间，递归更新右区间。

3. 否则，u 无法更新区间。

最后懒标记设为 v。

（u 和 v 在中点相等的情况也可以同样处理）

时间复杂度为 $O(m\log^2 n)$。

接下来考虑查询 x，直接在包含 x 的所有线段树区间的标记中比较答案即可。

接下来考虑查询 x，直接在包含 x 的所有线段树区间的标记中比较答案即可。

时间复杂度 $O(n\log n)$。

P4097 [HEOI2013]Segment

要求在平面直角坐标系下维护两个操作：

1. 在平面上加入一条线段。记第 i 条被插入的线段的标号为 i。

2. 给定一个数 k，询问与直线 $x=k$ 相交的线段中，交点纵坐标最大的线段的编号。$1 \le n \le 10^5$，$1 \le k, x_0, x_1 \le 39989$，$1 \le y_0, y_1 \le 10^9$。

P4097 [HEOI2013]Segment

要求在平面直角坐标系下维护两个操作：

1. 在平面上加入一条线段。记第 i 条被插入的线段的标号为 i。

2. 给定一个数 k，询问与直线 $x = k$ 相交的线段中，交点纵坐标最大的线段的编号。$1 \leq n \leq 10^5$，$1 \leq k, x_0, x_1 \leq 39989$，$1 \leq y_0, y_1 \leq 10^9$。

板子题。

P4655 [CEOI2017] Building Bridges

有 n 根柱子依次排列，每根柱子都有一个高度。第 i 根柱子的高度为 h_i。

现在想要建造若干座桥，如果一座桥架在第 i 根柱子和第 j 根柱子之间，那么需要 $(h_i - h_j)^2$ 的代价。

在造桥前，所有用不到的柱子都会被拆除，因为他们会干扰造桥进程。第 i 根柱子被拆除的代价为 w_i，注意 w_i 不一定非负，因为可能政府希望拆除某些柱子。

现在政府想要知道，通过桥梁把第 1 根柱子和第 n 根柱子连接的最小代价。注意桥梁不能在端点以外的任何地方相交。

$2 \leq n \leq 10^5, 0 \leq h_i, |w_i| \leq 10^6$

P4655 [CEOI2017] Building Bridges

首先，令 $s_i=\sum_{j=1}^{i} w_j$，f_i 为 1 到 i 连通的最小代价，有 dp 转移方程：

$$
\begin{aligned}
f_i &= \min\{f_j + (s_{i-1} - s_j) + (h_i - h_j)^2\} \\
&= \min\{(-2h_ih_j + f_j - s_j + h_j^2) + (s_{i-1} + h_i^2)\}
\end{aligned}
$$

P4655 [CEOI2017] Building Bridges

首先，令 $s_i = \sum_{j=1}^{i} w_j$，f_i 为 1 到 i 连通的最小代价，有 dp 转移方程：

$$
\begin{aligned}
f_i &= \min\{f_j + (s_{i-1} - s_j) + (h_i - h_j)^2\} \\
&= \min\{(-2h_ih_j + f_j - s_j + h_j^2) + (s_{i-1} + h_i^2)\}
\end{aligned}
$$

发现与 j 有关的项就是 $\min\{kx + b\}$ 的形式，直接用李超线段树维护。

P4655 [CEOI2017] Building Bridges

首先，令 $s_i=\sum_{j=1}^{i} w_j$，f_i 为 1 到 i 连通的最小代价，有 dp 转移方程：

$$
\begin{aligned}
f_i &= \min\{f_j+(s_{i-1}-s_j)+(h_i-h_j)^2\}\\
&= \min\{(-2h_ih_j+f_j-s_j+h_j^2)+(s_{i-1}+h_i^2)\}
\end{aligned}
$$

发现与 j 有关的项就是 $\min\{kx+b\}$ 的形式，直接用李超线段树维护。

因为都是直线，所以时间复杂度 $O(n\log n)$。

P4655 [CEOI2017] Building Bridges

首先，令 $s_i=\sum_{j=1}^{i} w_j$，f_i 为 1 到 i 连通的最小代价，有 dp 转移方程：

$$\begin{aligned} f_i &= \min\{f_j+(s_{i-1}-s_j)+(h_i-h_j)^2\} \\ &= \min\{(-2h_ih_j+f_j-s_j+h_j^2)+(s_{i-1}+h_i^2)\} \end{aligned}$$

发现与 j 有关的项就是 $\min\{kx+b\}$ 的形式，直接用李超线段树维护。

因为都是直线，所以时间复杂度 $O(n\log n)$。

（还可以用斜率优化做，但也是 $O(n\log n)$ 的）

CF932F Escape Through Leaf

有一颗 n 个节点的树（节点从 1 到 n 依次编号）。每个节点有两个权值，第 i 个节点的权值为 a_i, b_i。

你可以从一个节点跳到它的子树内任意一个节点上。从节点 x 跳到节点 y 一次的花费为 $a_x \times b_y$。跳跃多次走过一条路径的总费用为每次跳跃的费用之和。请分别计算出每个节点到达树的每个叶子节点的费用中的最小值。

注意：就算树的深度为 1，根节点也不算做叶子节点。另外，不能从一个节点跳到它自己.

$2 \le n \le 10^5$，$-10^5 \le a_i \le 10^5$，$-10^5 \le b_i \le 10^5$。

CF932F Escape Through Leaf

设 f_i 为 i 号节点的答案，有转移方程：
$f_i = \min\{f_j + a_i \times b_j\}$，其中 j 在 i 的子树中。

CF932F Escape Through Leaf

设 f_i 为 i 号节点的答案，有转移方程：
$f_i = \min\{f_j + a_i \times b_j\}$，其中 j 在 i 的子树中。
容易发现转移方程是 $\min\{kx + b\}$ 形式的，所以可以用李超线段树优化。

CF932F Escape Through Leaf

设 f_i 为 i 号节点的答案，有转移方程：

$f_i = \min\{f_j + a_i \times b_j\}$，其中 j 在 i 的子树中。

容易发现转移方程是 $\min\{kx + b\}$ 形式的，所以可以用李超线段树优化。

但是在树上转移，所以需要线段树合并。

CF932F Escape Through Leaf

现在只要考虑 2 棵李超线段树 a, b 的合并，其实只要用 b 当前区间的懒标记更新 a 的当前区间就行了，大体和线段树合并是一样的。

CF932F Escape Through Leaf

现在只要考虑 2 棵李超线段树 a, b 的合并，其实只要用 b 当前区间的懒标记更新 a 的当前区间就行了，大体和线段树合并是一样的。

这道题时间复杂度是 $O(n \log n)$ 的，具体可以势能分析。

区间 mex

给出长为 n 的序列 a，m 次询问，每次查询 $mex(a_l, \ldots, a_r)$。
$n, m \leq 5 \times 10^5$

区间 mex

对每一个区间 $[1, i]$ 建立一棵权值线段树，维护它们出现的最右边位置。

区间 mex

对每一个区间 $[1, i]$ 建立一棵权值线段树，维护它们出现的最右边位置。

查询即可在第 r 棵线段树上二分小于 l 的第一个位置即可，用可持久化线段树做。

CF1380F Strange Addition

对于非负整数 a, b，定义它们的" 奇异加法 $\oplus$" 如下：

1. 将 a, b 一上一下写在两行，并按照低位对齐。

2. 将它们的每一位加起来，并将每一位的结果从高位到低位连接在一起，得到的就是 $a \oplus b$。

（你可以认为，两个数都有无穷多的前导 0）

你现在有一个仅包含 n 个 $0 \sim 9$ 的数码的字符串 c，并且还有 m 次操作。一次操作为：

x d ：将 c 的第 x 个数码替换成数码 d 。

（注意，c 在任何时刻都可能存在前导 0）

每次操作过后，你需要输出，有多少个有序对 (a, b)，满足 a, b 都是非负整数，且 $a \oplus b = c$。

答案对 998244353 取模。

CF1380F Strange Addition

考虑对一个 c 串统计答案，设 f_i 表示 $1-i$ 的答案，转移如下 $f_i=(a_i+1)f_{i-1}+[a_{i-1}=1](9-a_i)f_{i-2}$。

CF1380F Strange Addition

考虑对一个 c 串统计答案，设 f_i 表示 $1-i$ 的答案，转移如下 $f_i=(a_i+1)f_{i-1}+[a_{i-1}=1](9-a_i)f_{i-2}$。
再把转移用矩阵形式表示，有

$$\begin{bmatrix} f_{i-2} & f_{i-1} \end{bmatrix} \times \begin{bmatrix} 0 & [a_{i-1}=1](9-a_i) \\ 1 & (a_i+1) \end{bmatrix} = \begin{bmatrix} f_{i-1} & f_i \end{bmatrix}$$

CF1380F Strange Addition

考虑对一个 c 串统计答案，设 f_i 表示 $1-i$ 的答案，转移如下
$f_i=(a_i+1)f_{i-1}+[a_{i-1}=1](9-a_i)f_{i-2}$。
再把转移用矩阵形式表示，有

$$\begin{bmatrix} f_{i-2} & f_{i-1} \end{bmatrix} \times \begin{bmatrix} 0 & [a_{i-1}=1](9-a_i) \\ 1 & (a_i+1) \end{bmatrix} = \begin{bmatrix} f_{i-1} & f_i \end{bmatrix}$$

所以只需要在线段树上维护区间上矩阵的乘积即可。
注意：边界条件和修改 a_x 的时候要修改 x 和 $x+1$ 位置的矩阵。

CF1380F Strange Addition

考虑对一个 c 串统计答案，设 f_i 表示 $1-i$ 的答案，转移如下 $f_i=(a_i+1)f_{i-1}+[a_{i-1}=1](9-a_i)f_{i-2}$。
再把转移用矩阵形式表示，有

$$\begin{bmatrix} f_{i-2} & f_{i-1} \end{bmatrix} \times \begin{bmatrix} 0 & [a_{i-1}=1](9-a_i) \\ 1 & (a_i+1) \end{bmatrix} = \begin{bmatrix} f_{i-1} & f_i \end{bmatrix}$$

所以只需要在线段树上维护区间上矩阵的乘积即可。
注意：边界条件和修改 a_x 的时候要修改 x 和 $x+1$ 位置的矩阵。
时间复杂度：$O(2^3(m+n)\log n)$。

luoguP4197 Peaks

给定一张 n 个点，m 条边的图，边权 c_i，点权 h_i。

q 次询问，每次询问从点 v 开始只经过边权小于等于 x 的边能到达的点中点权第 k 大的。

$n \le 10^5, m, q \le 5 \times 10^5, h_i, c_i, x \le 10^9$

luoguP4197 Peaks

离线做法：

将询问离线，按 x 和边权从小到大排序。考虑从小到大加入边，使得每次询问时，将所有边权小于等于 x 的边都加入图中。那么询问转化为 v 所在的连通块中第 k 大的。

luoguP4197 Peaks

离线做法：

将询问离线，按 x 和边权从小到大排序。考虑从小到大加入边，使得每次询问时，将所有边权小于等于 x 的边都加入图中。那么询问转化为 v 所在的连通块中第 k 大的。

用并查集维护连通快，同时对每一个连通块开一棵权值线段树维护第 k 大的询问。那么并查集合并 2 个连通块的时候，只要线段树合并就行了。

luoguP4197 Peaks

离线做法：

将询问离线，按 x 和边权从小到大排序。考虑从小到大加入边，使得每次询问时，将所有边权小于等于 x 的边都加入图中。那么询问转化为 v 所在的连通块中第 k 大的。

用并查集维护连通快，同时对每一个连通块开一棵权值线段树维护第 k 大的询问。那么并查集合并 2 个连通块的时候，只要线段树合并就行了。

时间复杂度：$O((n+q)\log n)$。

luoguP4197 Peaks

在线做法：

luoguP4197 Peaks

在线做法：

看到经过边权小于等于 x 的边，想到 $kruskal$ 重构树。于是建出 $kruskal$ 重构树，性质为原图中两个点之间的所有简单路径上最大边权的最小值 $=$ 重构树上两点 lca 的点权。

luoguP4197 Peaks

在线做法：

看到经过边权小于等于 x 的边，想到 $kruskal$ 重构树。于是建出 $kruskal$ 重构树，性质为原图中两个点之间的所有简单路径上最大边权的最小值 = 重构树上两点 lca 的点权。

于是询问时，只要找到 v 最近的祖先 u 满足点权小于等于 x。那 v 能走到的点集即为 u 子树中的所有点。

luoguP4197 Peaks

在线做法：

看到经过边权小于等于 x 的边，想到 $kruskal$ 重构树。于是建出 $kruskal$ 重构树，性质为原图中两个点之间的所有简单路径上最大边权的最小值 = 重构树上两点 lca 的点权。

于是询问时，只要找到 v 最近的祖先 u 满足点权小于等于 x。那 v 能走到的点集即为 u 子树中的所有点。

可以发现 u 子树中的点对应 dfs 序上的一段区间，所以预处理后变为求区间第 k 大。可持久化线段树可以解决。

luoguP4197 Peaks

在线做法：

看到经过边权小于等于 x 的边，想到 $kruskal$ 重构树。于是建出 $kruskal$ 重构树，性质为原图中两个点之间的所有简单路径上最大边权的最小值 = 重构树上两点 lca 的点权。

于是询问时，只要找到 v 最近的祖先 u 满足点权小于等于 x。那 v 能走到的点集即为 u 子树中的所有点。

可以发现 u 子树中的点对应 dfs 序上的一段区间，所以预处理后变为求区间第 k 大。可持久化线段树可以解决。

时间复杂度：$O((n+q)\log n)$。

luoguP4899 [IOI2018] werewolf 狼人

给出一张 n 个点，m 条边的图。Q 次询问，每次询问是否存在一条 (s,t) 的路径，满足先只走编号大于等于 l 的点，再只走编号小于等于 r 的点。

$n, Q \leq 2 \times 10^5, n-1 \leq m \leq 4 \times 10^5, s \neq t$

luoguP4899 [IOI2018] werewolf 狼人

看到编号都小于或大于某个值，再次用 *kruskal* 重构树。因为有小于和大于，所以建 2 棵，点权 a_i 分别为边的端点编号的 min 和 max。

luoguP4899 [IOI2018] werewolf 狼人

看到编号都小于或大于某个值，再次用 $kruskal$ 重构树。因为有小于和大于，所以建 2 棵，点权 a_i 分别为边的端点编号的 min 和 max。

比如考虑只走编号大于等于 l 的部分，就是先找到 s 深度最小的祖先 u 满足 $a_u \geq l$。那么 u 的子树中的点都是 s 可以走到的。

luoguP4899 [IOI2018] werewolf 狼人

看到编号都小于或大于某个值，再次用 $kruskal$ 重构树。因为有小于和大于，所以建 2 棵，点权 a_i 分别为边的端点编号的 min 和 max。

比如考虑只走编号大于等于 l 的部分，就是先找到 s 深度最小的祖先 u 满足 $a_u \geq l$。那么 u 的子树中的点都是 s 可以走到的。

假设 t 类似方法求出的祖先为 v。那么存在路径等价于 u 的子树中的点与 v 的子树中的点有重复。

luoguP4899 [IOI2018] werewolf 狼人

用上一题的方法将子树中的点转化为一段 dfs 序区间，那么问题转化为有 2 个排列 p, q，每次询问 $p_l, \ldots, p_r$ 和 $q_x, \ldots, q_y$ 是否有相同元素。

luoguP4899 [IOI2018] werewolf 狼人

用上一题的方法将子树中的点转化为一段 dfs 序区间，那么问题转化为有 2 个排列 p, q，每次询问 $p_l, \ldots, p_r$ 和 $q_x, \ldots, q_y$ 是否有相同元素。

令 pos_i 为 q_i 在 p 中出现的位置，那么询问转化为 $pos_x, \ldots, pos_y$ 是否有元素在区间 $[l, r]$ 中。用可持久化的权值线段树即可。

luoguP4899 [IOI2018] werewolf 狼人

用上一题的方法将子树中的点转化为一段 dfs 序区间，那么问题转化为有 2 个排列 p, q，每次询问 $p_l, \dots, p_r$ 和 $q_x, \dots, q_y$ 是否有相同元素。

令 pos_i 为 q_i 在 p 中出现的位置，那么询问转化为 $pos_x, \dots, pos_y$ 是否有元素在区间 $[l, r]$ 中。用可持久化的权值线段树即可。

时间复杂度：$O((n+q)\log n + m\log m)$。

CF765F Souvenirs

给出 n 以及一个长为 n 的序列 a。

给出 m，接下来 m 组询问。

每组询问给出一个 l, r，你需要求出，对于 $i, j \in [l, r]$，且满足 $i \neq j$，$|a_i - a_j|$ 的最小值。

$1 \leq n \leq 10^5$，$1 \leq m \leq 3 \times 10^5$，$0 \leq a_i \leq 10^9$。

CF765F Souvenirs

离线询问，考虑对每一个右端点 i 维护当前答案的后缀 $\min$。

CF765F Souvenirs

离线询问，考虑对每一个右端点 i 维护当前答案的后缀 min。

固定右端点 i，先只考虑满足 $j < i, a_j > a_i$ 的答案（值域翻转后就是其他的情况）。可以用值域线段树找出最后一个满足的 j，然后一个一个往前跳更新答案。

CF765F Souvenirs

离线询问，考虑对每一个右端点 i 维护当前答案的后缀 min。

固定右端点 i，先只考虑满足 $j < i, a_j > a_i$ 的答案（值域翻转后就是其他的情况）。可以用值域线段树找出最后一个满足的 j，然后一个一个往前跳更新答案。

但这太慢了。考虑当前位置为 j，如果下一次跳到 k 能更新答案，那么 $a_k - a_i < a_j - a_k$，即 $a_k - a_i < \frac{1}{2}(a_j - a_i)$。于是发现差值每次减半，最多跳 $O(log V)$ 个有效位置。

CF765F Souvenirs

离线询问，考虑对每一个右端点 i 维护当前答案的后缀 min。

固定右端点 i，先只考虑满足 $j < i, a_j > a_i$ 的答案（值域翻转后就是其他的情况）。可以用值域线段树找出最后一个满足的 j，然后一个一个往前跳更新答案。

但这太慢了。考虑当前位置为 j，如果下一次跳到 k 能更新答案，那么 $a_k - a_i < a_j - a_k$，即 $a_k - a_i < \frac{1}{2}(a_j - a_i)$。于是发现差值每次减半，最多跳 $O(log V)$ 个有效位置。

于是用权值线段树找有效位置，树状数组维护当前答案的后缀 min，时间复杂度 $O(n \log^2 V)$。

CF121E Lucky Array

定义“幸运数”为十进制下只有 4 和 7 的数。
给出长为 n 的数组 a，m 次操作。
1. 区间加 x
2. 查询区间中幸运数的个数。
$1 \le n, m \le 10^5, a_i \le 10^4$

CF121E Lucky Array

由于值域只有 10^4，容易发现幸运数只有 30 个。

CF121E Lucky Array

由于值域只有 10^4，容易发现幸运数只有 30 个。

将数域按幸运数分段，线段树维护区间中每个数到它右边第一个断点的距离的最小值及其个数。区间加就是区间减，在距离 <0 时暴力跳到下一个段。查询就是距离 $=0$ 的个数。

CF121E Lucky Array

由于值域只有 10^4，容易发现幸运数只有 30 个。

将数域按幸运数分段，线段树维护区间中每个数到它右边第一个断点的距离的最小值及其个数。区间加就是区间减，在距离 <0 时暴力跳到下一个段。查询就是距离 $=0$ 的个数。

时间复杂度：$O(30n\log n)$。

luoguP1712 [NOI2016] 区间

给出 n 个区间，你需要选出 m 个区间，满足它们至少包含一个共同点，最小化这 m 个区间的最大长度 − 最小长度。

$1 \le n \le 5 \times 10^5, 1 \le m \le 2 \times 10^5$

luoguP1712 [NOI2016] 区间

按长度排序，双指针扫求答案。用线段树维护每一个点被覆盖了几次，只要支持区间加，区间减和全局最大值。

luoguP1712 [NOI2016] 区间

按长度排序，双指针扫求答案。用线段树维护每一个点被覆盖了几次，只要支持区间加，区间减和全局最大值。

离散化缩小值域，时间复杂度 $O(n\log n)$。

CF678F Lena and Queries

n 个操作:

1. 添加一对 (x, y) 进入集合。

2. 删除第 k 个操作所添加的数对，保证第 k 个操作是操作 1. 且对应的数对未被删除。

3. 给定参数 k 求集合中 $kx + y$ 的最大值。

值域 $[-10^9, 19^9]$

CF678F Lena and Queries

首先维护上凸包可以通过二分斜率求出 $kx + y$ 的最大值。

CF678F Lena and Queries

首先维护上凸包可以通过二分斜率求出 $kx+y$ 的最大值。

线段树分治，每个线段树节点建立一个上凸包，暴力二分斜率可以做到 $O(n\log n\log V)$。

CF678F Lena and Queries

首先维护上凸包可以通过二分斜率求出 $kx+y$ 的最大值。

线段树分治，每个线段树节点建立一个上凸包，暴力二分斜率可以做到 $O(n\log n\log V)$。

发现总共就插入 $O(n)$ 个点，所以可以先排序好再一个一个插入。于是建凸包部分只有 $O(n\log n)$。

对查询类似，先排好序，然后贪心地再凸包上找答案。于是查询部分也只有 $O(n\log n)$。总时间复杂度 $O(n\log n)$。

luoguP4198 楼房重建

有 n 栋楼房，第 i 栋可以表示成线段 $(i,0)->(i,h_i)$。如果这栋楼上任何一个高度与原点连线没有与之前的楼房相交，就称这栋楼是可见的。

楼房初始高度为 0，m 次操作，每次将横坐标为 x 的楼高度改为 y。每次修改后，询问从原点能看到多少栋楼。

$1 \le n, m \le 10^5$

luoguP4198 楼房重建

令 $k_i = \frac{h_i}{i}$，则相当于每次跳到后面第一个 k 更大的。

luoguP4198 楼房重建

令 $k_i = \frac{h_i}{i}$，则相当于每次跳到后面第一个 k 更大的。

下面考虑的都是 k_i，线段树维护区间 max 和单独拎出来时的答案 res。考虑如何对答案 $push\_up$，假设当前节点为 p。

luoguP4198 楼房重建

令 $k_i = \frac{h_i}{i}$，则相当于每次跳到后面第一个 k 更大的。

下面考虑的都是 k_i，线段树维护区间 max 和单独拎出来时的答案 res。考虑如何对答案 $push\_up$，假设当前节点为 p。

ls 的答案显然是 p 的答案，对于右儿子递归计算。令 $s(p,x)$ 为递归到节点 p，前面的 max 为 x 时的答案。

luoguP4198 楼房重建

令 $k_i = \frac{h_i}{i}$，则相当于每次跳到后面第一个 k 更大的。

下面考虑的都是 k_i，线段树维护区间 max 和单独拎出来时的答案 res。考虑如何对答案 $push\_up$，假设当前节点为 p。

ls 的答案显然是 p 的答案，对于右儿子递归计算。令 $s(p, x)$ 为递归到节点 p，前面的 max 为 x 时的答案。

若左儿子的最大值小于等于 x，那么直接递归到右儿子。

否则递归到左儿子，加上 $res_p - res_{ls}$。

luoguP4198 楼房重建

令 $k_i = \frac{h_i}{i}$，则相当于每次跳到后面第一个 k 更大的。

下面考虑的都是 k_i，线段树维护区间 max 和单独拎出来时的答案 res。考虑如何对答案 $push\_up$，假设当前节点为 p。

ls 的答案显然是 p 的答案，对于右儿子递归计算。令 $s(p, x)$ 为递归到节点 p，前面的 max 为 x 时的答案。

若左儿子的最大值小于等于 x，那么直接递归到右儿子。

否则递归到左儿子，加上 $res_p - res_{ls}$。

这与李超线段树的做法是类似的，即先定位到影响区间，然后通过一些性质使得暴力修改时只递归一侧的子树，时间复杂度为 $O(n \log^2 n)$。

51nod 1766 树上的最远点对

给定一棵 n 个点的树，m 次询问，每次询问在区间 $[a, b], [c, d]$ 中各选一个点，使这两点距离最大。

$1 \leq n, m \leq 10^5$

51nod 1766 树上的最远点对

首先有一个直径的性质：若点集 S_1 的直径端点为 u_1, u_2，S_2 的直径端点为 v_1, v_2，且 $S_1 \cap S_2 = \varnothing$，那么 $S_1 \cup S_2$ 的直径端点一定是 u_1, u_2, v_1, v_2 中的两个。

51nod 1766 树上的最远点对

首先有一个直径的性质：若点集 S_1 的直径端点为 u_1, u_2，S_2 的直径端点为 v_1, v_2，且 $S_1 \cap S_2 = \varnothing$，那么 $S_1 \cup S_2$ 的直径端点一定是 u_1, u_2, v_1, v_2 中的两个。

于是线段树维护区间中点的直径端点，可以简单合并左右儿子。查询也可以简单的做了。因为要查询很多次距离，所以可以用欧拉序和 RMQ 来做。

51nod 1766 树上的最远点对

首先有一个直径的性质：若点集 S_1 的直径端点为 u_1, u_2，S_2 的直径端点为 v_1, v_2，且 $S_1 \cap S_2 = \varnothing$，那么 $S_1 \cup S_2$ 的直径端点一定是 u_1, u_2, v_1, v_2 中的两个。

于是线段树维护区间中点的直径端点，可以简单合并左右儿子。查询也可以简单的做了。因为要查询很多次距离，所以可以用欧拉序和 RMQ 来做。

时间复杂度：$O(n \log n)$。