

主席树进阶

Cpt.

定义 \Rightarrow

定义=> ~~然而我也没查到百度百科~~

定义=> ~~然而我也没查到百度百科~~

主席树就是利用函数式编程的思想来使线段树支持询问历史版本、同时充分利用它们之间的共同数据来减少时间和空间消耗的增强版的线段树。

主席树的空间复杂度可以达到

$O(N \log N)$ (无修改的情况下)。

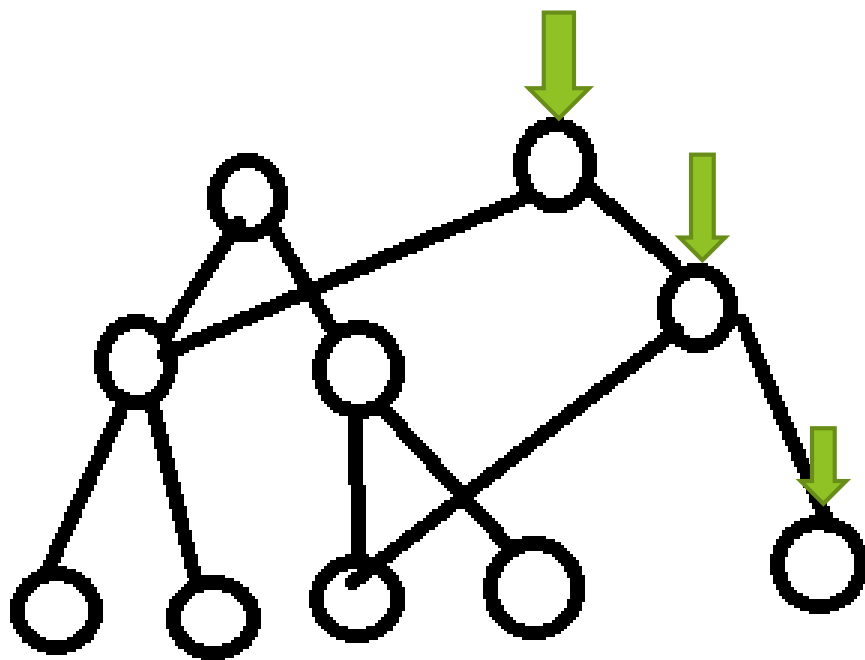
因为每一次修改至多修改 $\log N$ 个点。

定义 \Rightarrow

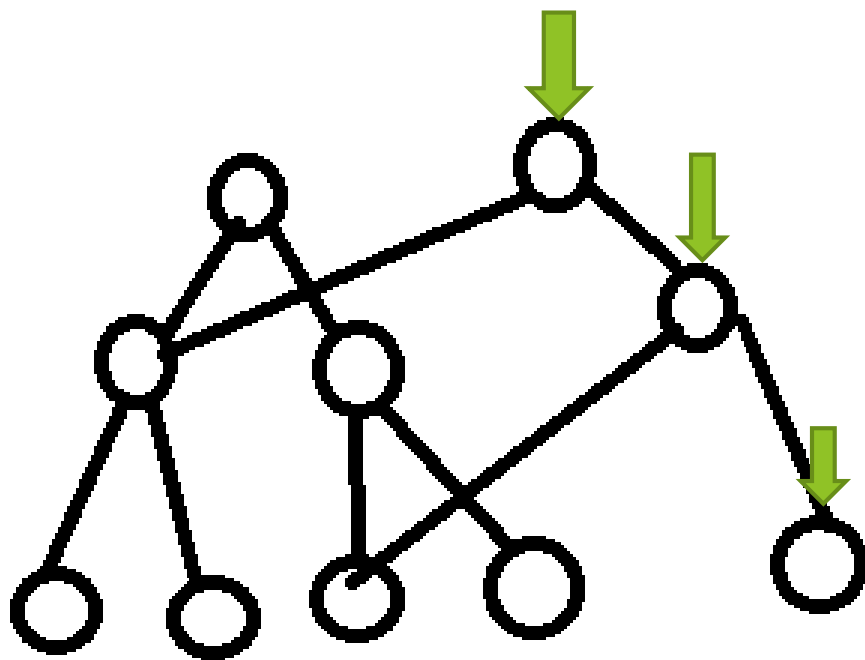
定义 \Rightarrow

所谓主席树呢，就是对原来的数列 $[1..n]$ 的每一个前缀 $[1..i]$
($1 \leq i \leq n$) 建立一棵线段树。
所以对于查询区间 $(x \sim y)$ ，只需要
访问 $\text{Segtree}[x-1]$ 和 $\text{Segtree}[y]$
就好了。

这差不多看做一颗简单的主席树



这差不多看做一颗简单的主席树



绿色箭头为被修改的节点，需要另外建点，
每次最多修改 $\log N$ 个节点，这保证了时空复杂度。

HDU2665

给定N个数，求区间第K大。

HDU2665

这道题明显可以用划分树做。但划分树是题外话。有兴趣的同学可以课后百度。那怎么用主席树做呢？

HDU2665

这道题明显可以用划分树做。但划分树是题外话。有兴趣的同学可以课后百度。那怎么用主席树做呢？

以权值为节点建一颗权值主席树。
所以我们要离散化。
然后对于一颗权值线段树的第 k 大查询比较简单，主席树操作差不多。

HDU2665

这道题明显可以用划分树做。但划分树是题外话。有兴趣的同学可以课后百度。那怎么用主席树做呢？

以权值为节点建一颗权值主席树。
所以我们要离散化。
然后对于一颗权值线段树的第k大查询比较简单，主席树操作差不多。

类似的模板题还有：ZOJ2112
POJ2104

HHH0J#44

可持久化书橱

HHHOJ#44

这道题相信大家不会陌生，是一道构造然后DFS的题目。

HHHOJ#44

这道题相信大家不会陌生，是一道构造然后DFS的题目。

因为我讲的是主席树专题，所以这次讲的是主席树写法。

HHHOJ#44

对于第一种操作，我们直接插入主席树中。

HHHOJ#44

对于第一种操作，我们直接插入主席树中。

对于第二种操作，也是直接插入。

HHHOJ#44

对于第一种操作，我们直接插入主席树中。

对于第二种操作，也是直接插入。

对于第三种操作，类似区间修改，以区间异或实现，用Lazy标记翻转状态。

HHHOJ#44

对于第一种操作，我们直接插入主席树中。

对于第二种操作，也是直接插入。

对于第三种操作，类似区间修改，以区间异或实现，用Lazy标记翻转状态。

对于第四种操作，最易实现，Rt赋值为要回到的那个状态就好了。

小Z是一个小有名气的钢琴家，最近C博士送给了小Z一架超级钢琴，小Z希望能够用这架钢琴创作出世界上最美妙的音乐。这架超级钢琴可以弹奏出 n 个音符，编号为1至 n 。第 i 个音符的美妙度为 A_i ，其中 A_i 可正可负。一个“超级和弦”由若干个编号连续的音符组成，包含的音符个数不少于 L 且不多于 R 。我们定义超级和弦的美妙度为其包含的所有音符的美妙度之和。两个超级和弦被认为是相同的，当且仅当这两个超级和弦所包含的音符集合是相同的。小Z决定创作一首由 k 个超级和弦组成的乐曲，为了使得乐曲更加动听，小Z要求该乐曲由 k 个不同的超级和弦组成。我们定义一首乐曲的美妙度为其所包含的所有超级和弦的美妙度之和。小Z想知道他能够创作出来的乐曲美妙度最大值是多少。

BZOJ2006

首先我们可以现将对于，每个 $i(i \geq L)$ ，预处理出可以选的区间里的最大值，并放到堆里。

BZOJ2006

首先我们可以现将对于，每个 $i(i \geq L)$ ，预处理出可以选的区间里的最大值，并放到堆里。

然后取出堆顶 K 次，取出后将取出的区间的下一个比它大的值放入堆中，否则Continue。

BZOJ2006

首先我们可以现将对于，每个 $i(i \geq L)$ ，预处理出可以选的区间里的最大值，并放到堆里。

然后取出堆顶 K 次，取出后将取出的区间的下一个比它大的值放入堆中，否则Continue。

这样我们就能得到最优答案

首先我们可以现将对于，每个 $i(i \geq L)$ ，预处理出可以选的区间里的最大值，并放到堆里。

然后取出堆顶 K 次，取出后将取出的区间的下一个比它大的值放入堆中，否则Continue。

这样我们就能得到最优答案

类似于取一个三元组 (c, r, k) ，表示值为 c ，区间以 r 结尾，在满足条件的 l 中前缀和为第 k 小的答案。

BZ0J3956

定义一个点对 (i, j) $i < j$, 满足以下条件之一则为好点对:

$$1. i = j - 1$$

$$2. \forall k \in (i, j), A_k < \min\{A_i, A_j\}$$

BZ0J3956

定义一个点对 (i, j) $i < j$, 满足以下条件之一则为好点对:

1 $i = j - 1$

2 $\forall k \in (i, j), A_k < \min\{A_i, A_j\}$

给定 N 个数, M 组询问, 每次询问 $[L, R]$ 有多少对好点对。

BZ0J3956

定义一个点对 (i, j) $i < j$, 满足以下条件之一则为好点对:

1 $i = j - 1$

2 $\forall k \in (i, j), A_k < \min\{A_i, A_j\}$

给定 N 个数, M 组询问, 每次询问 $[L, R]$ 有多少对好点对。

不强制在线怎么做?

BZ0J3956

定义一个点对 (i, j) $i < j$, 满足以下条件之一则为好点对:

1. $i = j - 1$

2. $\forall k \in (i, j), A_k < \min\{A_i, A_j\}$

给定 N 个数, M 组询问, 每次询问 $[L, R]$ 有多少对好点对。

不强制在线怎么做?

强制在线又怎么做?

BZ0J3956

首先我们可以从定义中得知，好点对最多只有 $2*N$ 对，所以预处理出每一对好点对。

BZ0J3956

首先我们可以从定义中得知，好点对最多只有 $2*N$ 对，所以预处理出每一对好点对。
怎么处理呢？

BZ0J3956

首先我们可以从定义中得知，好点对最多只有 $2*N$ 对，所以预处理出每一对好点对。

怎么处理呢？

我们需要一个强大的线性算法/数据结构来维护。

首先我们可以从定义中得知，好点对最多只有 $2*N$ 对，所以预处理出每一对好点对。

怎么处理呢？

我们需要一个强大的线性算法/数据结构来维护。

-----单调栈

BZ0J3956

处理出点对之后就很简单了。

BZ0J3956

处理出点对之后就很简单了。

先将点对按右端点插入到主席树里去，
直接询问区间就好了。