提高组数据结构

吉如一

杭州天水幼儿园

自我介绍

- NOIP2013-2015 提高组一等奖

自我介绍

- NOIP2013-2015 提高组一等奖
- NOI2014 银牌

自我介绍

- NOIP2013-2015 提高组一等奖
- NOI2014 银牌
- NOI2015 金牌

自我介绍

- NOIP2013-2015 提高组一等奖
- NOI2014 银牌
- NOI2015 金牌
- 目前就读于北京大学信息科学技术学院

自我介绍

- NOIP2013-2015 提高组一等奖
- NOI2014 银牌
- NOI2015 金牌
- 目前就读于北京大学信息科学技术学院
- jiruyi910387714@edu.pku.cn

基本框架

- 链表

基本框架

- 链表
- 栈与队列

基本框架

- 链表
- 栈与队列
- 二叉树

基本框架

- 链表
- 栈与队列
- 二叉树
- **树状数组**

基本框架

- 链表
- 栈与队列
- 二叉树
- **树状数组**
- **ST 表**

基本框架

- 链表
- 栈与队列
- 二叉树
- **树状数组**
- **ST 表**
- **堆**

基本框架

- 链表
- 栈与队列
- 二叉树
- **树状数组**
- **ST 表**
- **堆**
- 线段树

基本框架

- 链表
- 栈与队列
- 二叉树
- **树状数组**
- **ST 表**
- **堆**
- 线段树
- map 与 set

链表

- 单向链表、双向链表（插入删除）

链表

- 单向链表、双向链表（插入删除）
- 邻接表

链表

13. 双向链表中有两个指针域，llink 和 rlink，分别指回前驱及后继，设 p 指向链表中的一个结点，q 指向一待插入结点，现要求在 p 前插入 q，则正确的插入为（　　）。

A.
```
p->llink = q; q->rlink = p;
p->llink->rlink = q; q->llink = p->llink;
```

B.
```
q->llink = p->llink; p->llink->rlink = q;
q->rlink = p; p->llink = q->rlink;
```

C.
```
q->rlink = p; p->rlink = q;
p->llink->rlink = q; q->rlink = p;
```

D.
```
p->llink->rlink = q; q->rlink = p;
q->llink = p->llink; p->llink = q;
```

链表

1．如果有一个表从不修改，那么就可以使用一种更简单的方法来实现表的元素的查找。为了有效地访问第 i 个元素，向单链表的每个元素中添加第二个指针，使其指向表中其它元素来减少查找所需时间。

(1) 请设计这样的“超级跳表”数据结构，请写出查找第 i 个元素的伪代码；

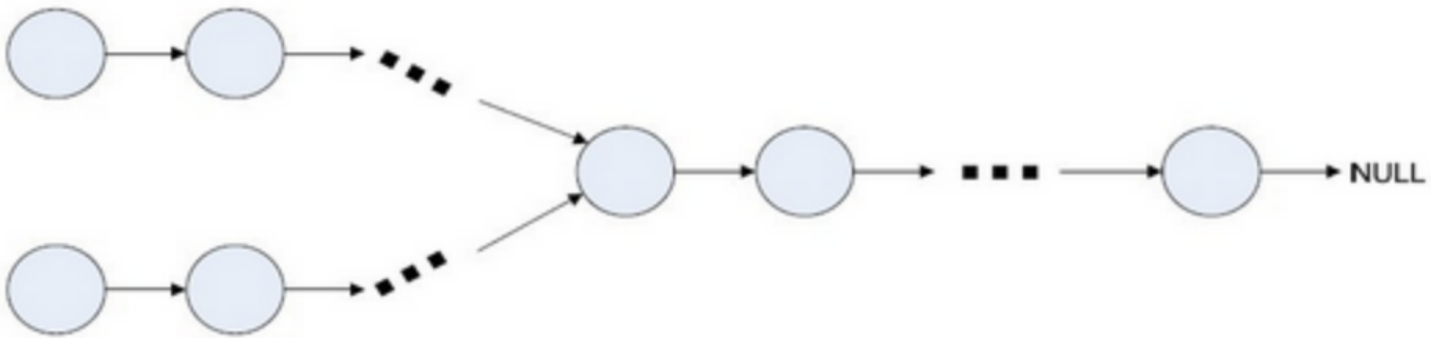
(2) 分析上述算法的时间代价，说明它是 O(log n) 时间的。

链表

3．已知两个单向链表 La 和 Lb，请设计算法解决如下问题，并分析算法的时间复杂度和空间复杂度。

(1) 判断 La 和 Lb 是否相交（下图所示为链表相交）

(2) 找出第一个相交的节点



链表

给出排列 A，对每一个 i 询问区间 $(i, n]$ 中大于 A_i 的最小数。

$n \leq 10^5$

栈

- 判断一个序列是否是合法的出栈序。

栈

- 判断一个序列是否是合法的出栈序。
- 求长度为 n 的排列本质不同的出栈序数目。

栈

- 判断一个序列是否是合法的出栈序。
- 求长度为 n 的排列本质不同的出栈序数目。
- 表达式求值。

面试题

在常数时间里实现一个栈的插入，删除，和返回最小值操作。

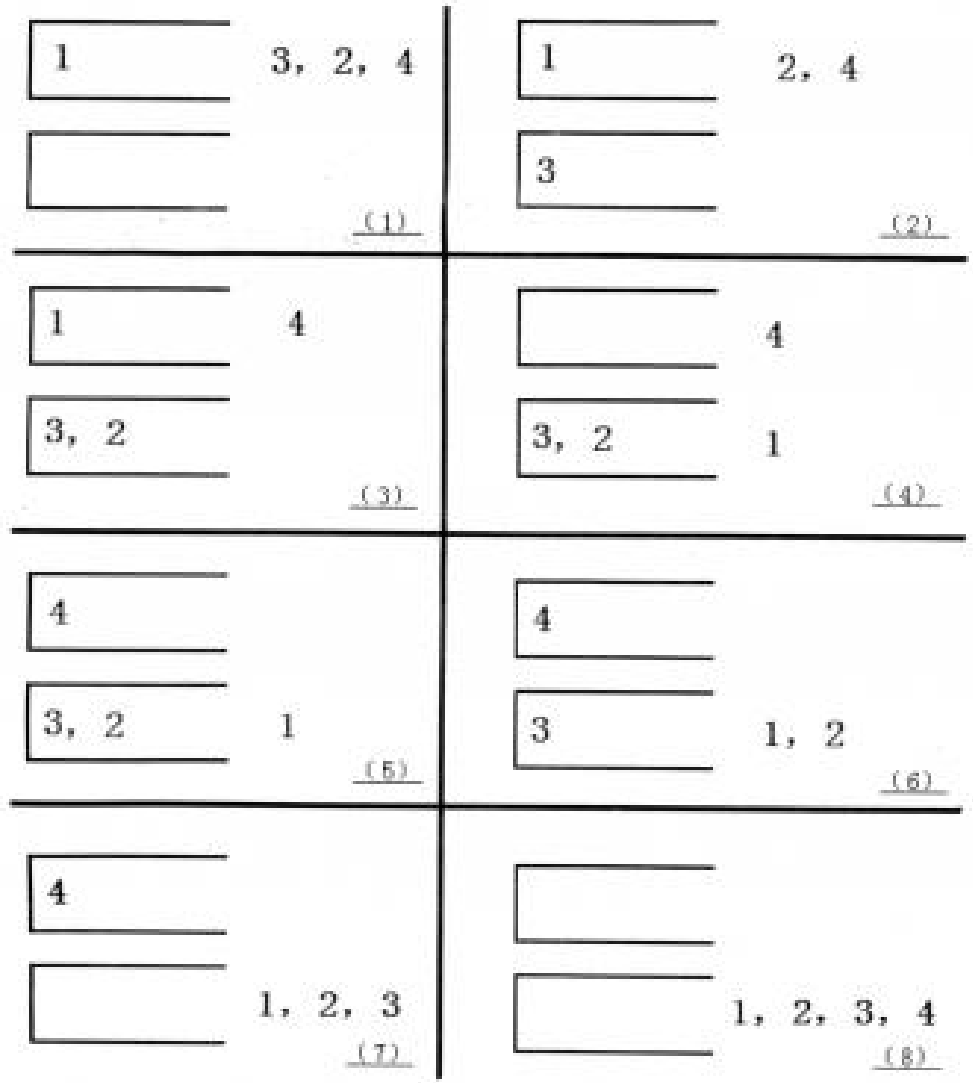
双栈排序

给定一个长度为 n 的排列 A，你有两个栈，每一时刻你可以向一个栈中压一个数或者弹出栈顶元素，要求弹出的数有序，求一个字典序最小的方案。

定义操作的大小关系：第一个栈入栈 $>$ 第一个栈出栈 $>$ 第二个栈入栈 $>$ 第二个栈出栈。

$n \leq 1000$

双栈排序



双栈排序

- i 和 j 不能进入同一个栈当且仅当存在 k 满足 $i < j < k$ 且 $A_k < A_i < A_j$。

双栈排序

- i 和 j 不能进入同一个栈当且仅当存在 k 满足 $i < j < k$ 且 $A_k < A_i < A_j$。
- 如果 i 和 j 不能进入同一个栈，就连一条边。

双栈排序

- i 和 j 不能进入同一个栈当且仅当存在 k 满足 $i < j < k$ 且 $A_k < A_i < A_j$。
- 如果 i 和 j 不能进入同一个栈，就连一条边。
- 二分图染色，模拟。

双栈排序

- i 和 j 不能进入同一个栈当且仅当存在 k 满足 $i < j < k$ 且 $A_k < A_i < A_j$。
- 如果 i 和 j 不能进入同一个栈，就连一条边。
- 二分图染色，模拟。
- 是不是任意的序列都能被双栈排序？

三栈排序

你有三个栈，最开始第一个栈中有 n 个数。

每次你可以选取两个栈 a 和 b，将 a 的栈顶元素移动到 b 的栈顶。

构造一个长度不超过 m 的操作序列使得最后所有数都在同一个栈中且这个栈有序。

$n \leq 10^4, m = 10^6$

队列

- BFS，SPFA

队列

- BFS，SPFA
- 循环队列的写法

面试题

用两个栈实现一个队列。

二叉树

- 满二叉树、完全二叉树（定义模糊）

二叉树

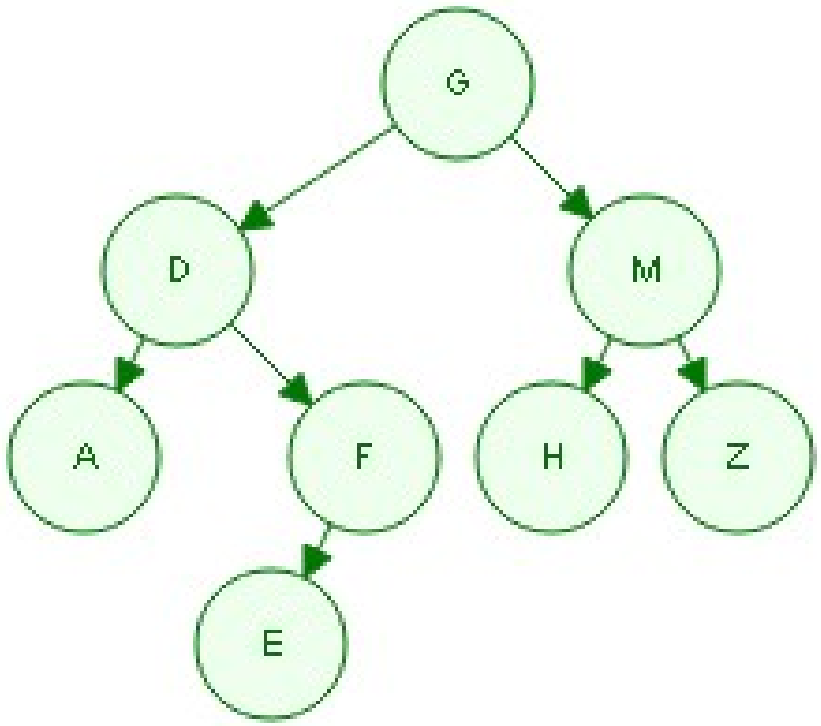
- 满二叉树、完全二叉树（定义模糊）
- n 个节点不同的有根二叉树数目（左右儿子不同）。

二叉树

- 前序遍历，中序遍历，后序遍历

二叉树

- 前序遍历，中序遍历，后序遍历



二叉树

- 前中求后

二叉树

- 前中求后
- 前序遍历：GDAFEMHZ

二叉树

- 前中求后
- 前序遍历：GDAFEMHZ
- 中序遍历：ADEFGHMZ

二叉树

- 前中求后
- 前序遍历：GDAFEMHZ
- 中序遍历：ADEFGHMZ
- 后序遍历：AEFDHZMG

树状数组

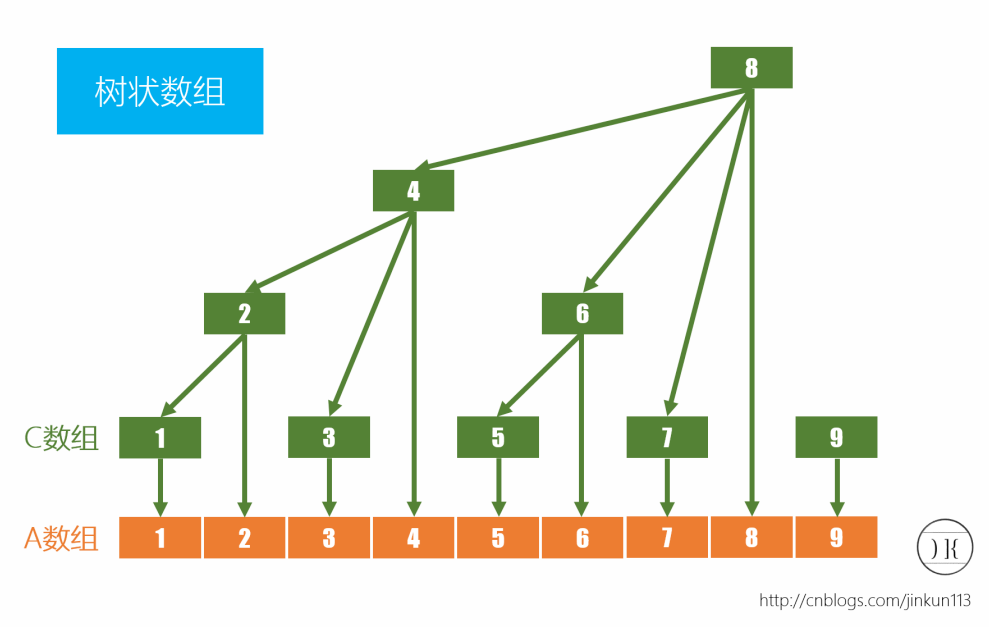
- 单点修改，询问区间信息

树状数组

- 单点修改，询问区间信息
- 核心：令 f_i 为 i 最小的为 1 的二进制位的值，让 A_i 维护 $(i - f_i, i]$ 的信息。

树状数组

- 单点修改，询问区间信息
- 核心：令 f_i 为 i 最小的为 1 的二进制位的值，让 A_i 维护 $(i - f_i, i]$ 的信息。



树状数组

- 考虑询问区间 $[1,i]$，$i \to i - f_i$

树状数组

- 考虑询问区间 $[1, i]$，$i \to i - f_i$
- 修改 i，$i \to i + f_i$

树状数组

- 考虑询问区间 $[1, i]$，$i \to i - f_i$
- 修改 i，$i \to i + f_i$
- 时间复杂度 $O(\log n)$

树状数组

- $f_{2i} = 2 \times f_i$

树状数组

- $f_{2i} = 2 \times f_i$
- $f_{2i+1} = 1$

树状数组

- $f_{2i} = 2 \times f_i$
- $f_{2i+1} = 1$
- $f_i = i$ and $(-i)$

树状数组

- $f_{2i} = 2 \times f_i$
- $f_{2i+1} = 1$
- $f_i = i$ and $(-i)$
- 负数用补码表示。

树状数组

- 逆序对。

树状数组

- 逆序对。
- LIS。

树状数组

- 逆序对。
- LIS。
- 单点修改，询问矩形和。

树状数组

- 逆序对。
- LIS。
- 单点修改，询问矩形和。
- 区间加，询问区间和。

树状数组

- 逆序对。
- LIS。
- 单点修改，询问矩形和。
- 区间加，询问区间和。
- 矩形加，询问矩形和。

ST 表

- 不带修改询问区间最小值。

ST 表

- 不带修改询问区间最小值。
- 令 f_i 为满足 $2^{f_i} \leq i$ 的最大整数。

ST 表

- 不带修改询问区间最小值。
- 令 f_i 为满足 $2^{f_i} \leq i$ 的最大整数。
- 预处理 $w[i][j]$ 为 $[i, i + 2^j)$ 的最小值

ST 表

- 不带修改询问区间最小值。
- 令 f_i 为满足 $2^{f_i} \leq i$ 的最大整数。
- 预处理 $w[i][j]$ 为 $[i, i+2^j)$ 的最小值
- $\min[l, r] = \min(w[l][f_{r-l+1}], w[r-2^{f_{r-l+1}}+1][f_{r-l+1}])$

ST 表

- 不带修改询问区间最小值。
- 令 f_i 为满足 $2^{f_i} \leq i$ 的最大整数。
- 预处理 $w[i][j]$ 为 $[i, i+2^j)$ 的最小值
- $\min[l, r] = \min(w[l][f_{r-l+1}], w[r-2^{f_{r-l+1}}+1][f_{r-l+1}])$
- $f_i = f_{\lfloor \frac{i}{2} \rfloor} + 1$

ST 表

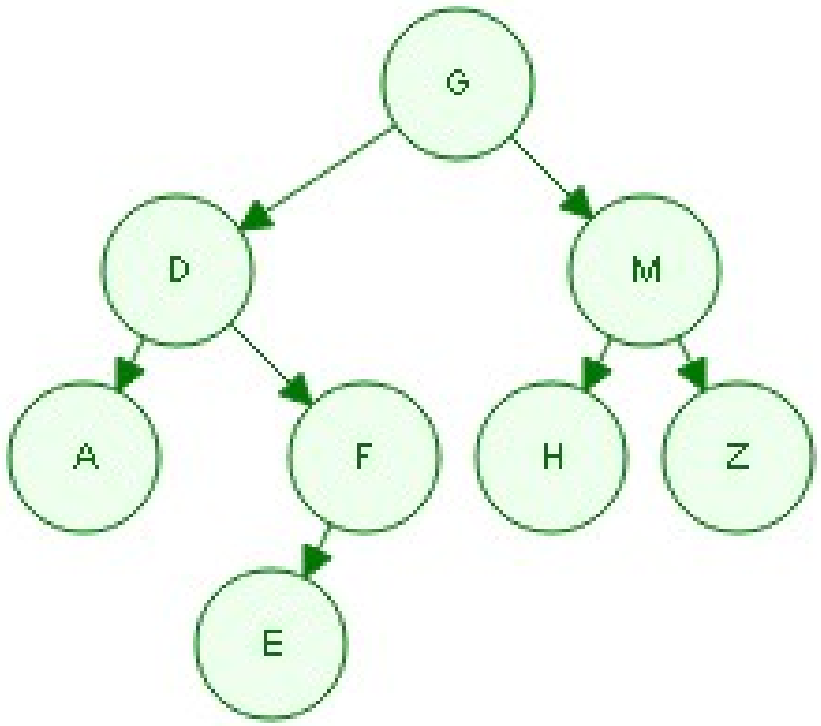
- 不带修改询问区间最小值。
- 令 f_i 为满足 $2^{f_i} \leq i$ 的最大整数。
- 预处理 $w[i][j]$ 为 $[i, i+2^j)$ 的最小值
- $\min[l, r] = \min(w[l][f_{r-l+1}], w[r-2^{f_{r-l+1}}+1][f_{r-l+1}])$
- $f_i = f_{\lfloor \frac{i}{2} \rfloor} + 1$
- $f_i = \lfloor \frac{\log(i)}{\log(2)} \rfloor$

ST 表

- 最近公共祖先

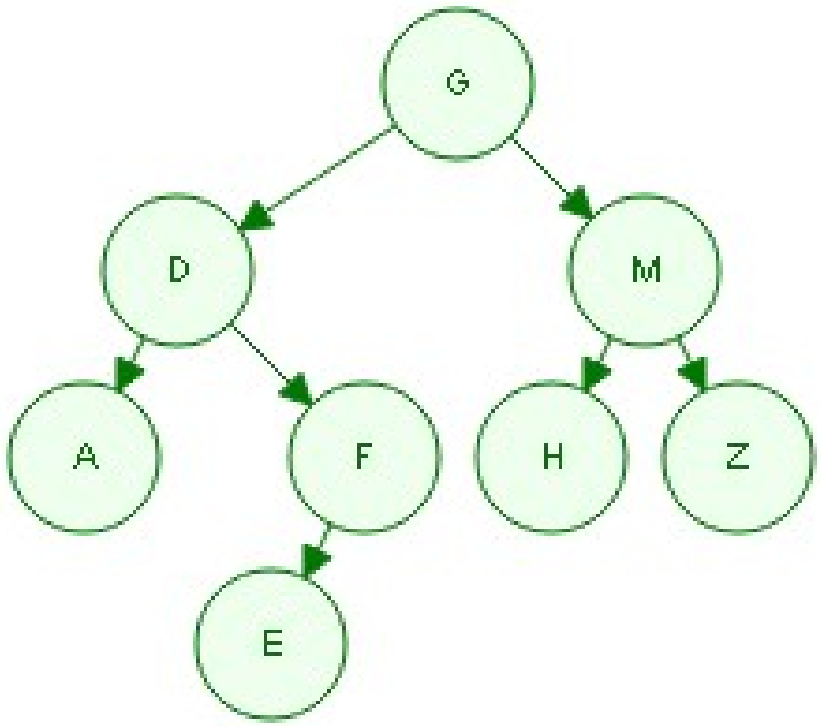
ST 表

- 最近公共祖先



ST 表

- 最近公共祖先



- {GDADFEFDGMHMZMG}

堆

- 插入、删除最小值、询问最小值。

堆

- 插入、删除最小值、询问最小值。
- priority_queue

堆

- 插入、删除最小值、询问最小值。
- priority_queue
- 手写推荐左偏树。

左偏树

- $w_i \leq w_{l_i}, w_i \leq w_{r_i}$

左偏树

- $w_i \leq w_{l_i}, w_i \leq w_{r_i}$
- 左子树深度不小于右子树深度

左偏树

- $w_i \leq w_{l_i}, w_i \leq w_{r_i}$
- 左子树深度不小于右子树深度
- 左子树和右子树都是左偏树

左偏树

- $w_i \leq w_{l_i}, w_i \leq w_{r_i}$
- 左子树深度不小于右子树深度
- 左子树和右子树都是左偏树
- **一直向右走O(log n)步就能到达叶子节点**

左偏树

```
struct tree{
    int l,r,w,d;
}t[N];
int merge(int k1,int k2){
    if (k1==0||k2==0) return k1+k2;
    if (t[k1].w>t[k2].w) swap(k1,k2);
    t[k1].r=merge(t[k1].r,k2);
    if (t[t[k1].l].d<t[t[k1].r].d) swap(t[k1].l,t[k1].r);
    t[k1].d=t[t[k1].r].d+1;
    return k1;
}
```

左偏树

- 插入：合并原树和新节点。

左偏树

- 插入：合并原树和新节点。
- 删除最小值：合并左右子树。

堆

- 哈夫曼树

堆

- 哈夫曼树
- Dijkstra

堆

- 哈夫曼树
- Dijkstra
- 常和贪心题一起出现

堆

有 n 个房间和 n 盏灯，你需要在每个房间里放入一盏灯。每盏灯都有一定功率，每间房间都需要不少于一定功率的灯泡才可以完全照亮。

你可以去附近的商店换新灯泡，商店里所有正整数功率的灯泡都有售。但由于背包空间有限，你至多只能换 k 个灯泡。

你需要找到一个合理的方案使得每个房间都被完全照亮，并在这个前提下使得总功率尽可能小。

$n \leq 5 \times 10^5$

线段树

- 区间操作

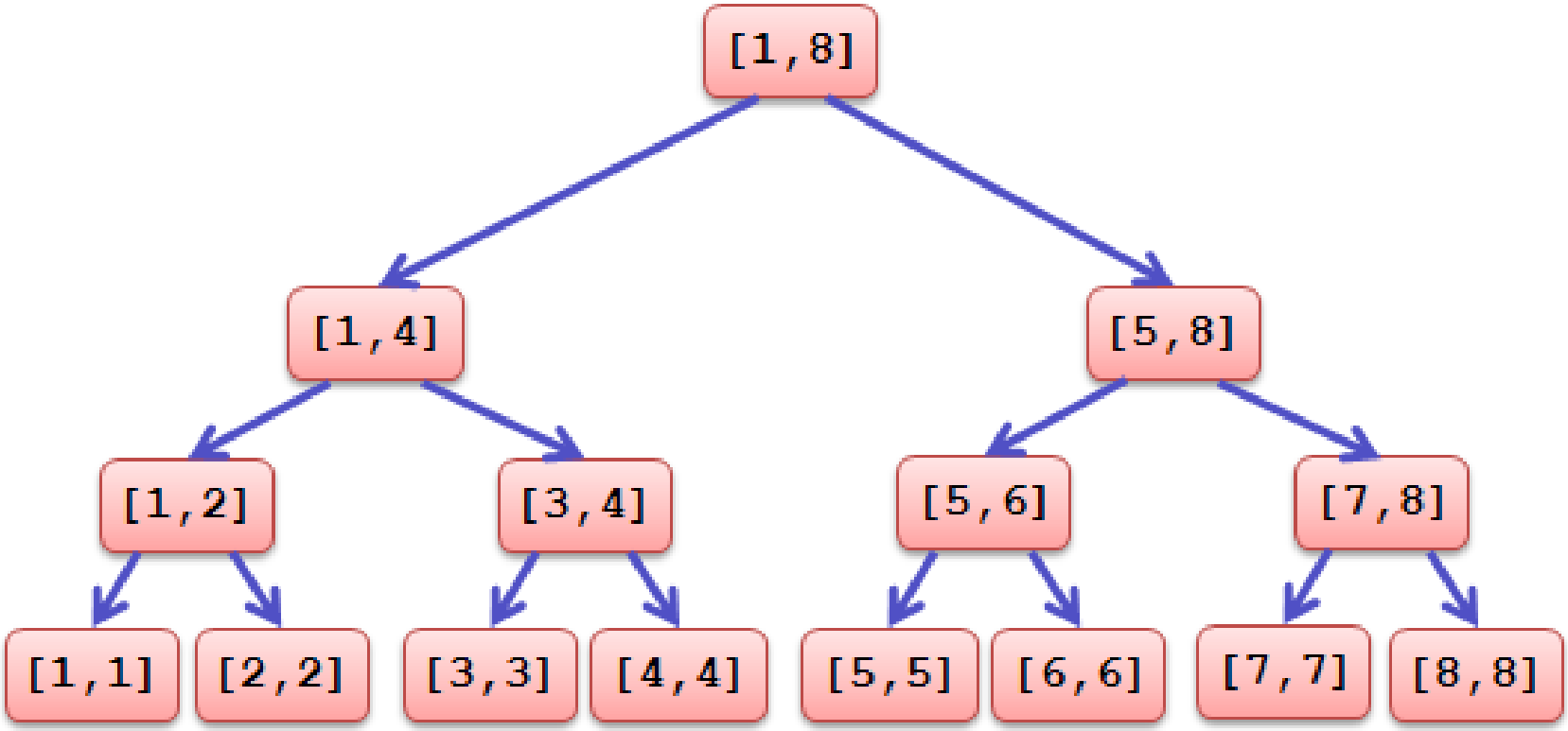
线段树

- 区间操作
- 核心：把任意区间拆成线段树上 $O(\log n)$ 个节点。

线段树

- 区间操作
- 核心：把任意区间拆成线段树上 $O(\log n)$ 个节点。
- 懒标记法：给节点打上标记，要访问其子节点的时候再把标记下传。

线段树



线段树

```
void ope(int whe,int l,int r,int L,int R){
    if (l>R||r<L) return;
    if (l>=L&&r<=R){
        ope(whe); return;
    }
    int mid=l+r>>1; pushdown(whe);
    ope(whe*2,l,mid,L,R);
    ope(whe*2+1,mid+1,r,L,R);
    upd(whe);
}
```

map 与 set

- set：插入删除找大于等于某一个数的第一个数。（有重复值时需使用 multiset）

map 与 set

- set：插入删除找大于等于某一个数的第一个数。（有重复值时需使用 multiset）
- map: 可以当数组用。

谢谢大家