数据结构

余行江

清华大学

2017 年 1 月 15 日

Table of Contents

并查集 (Disjoint-set Data Structure)

1. 处理不相交集合（Disjoint Sets）的合并及查询问题
   - 合并两个集合
   - 查询某个元素所属的集合
2. 合并 $\Rightarrow$ 连边，所属 $\Rightarrow$ 联通块
3. 用树结构维护联通块

路径压缩

1. 每次查询树上一条路径后将每个点直接指向树根

按秩合并 (union by rank)

1. OR (启发式合并)
2. 每次合并时将元素个数少的集合指向元素个数多的集合
3. $\Rightarrow$ 每次向父亲走 子树大小 $\times 2$

优先队列 (Priority Queue)

1. 维护集合的 max priority（例如最大值，最小值）
   - 添加 / 删除元素

二叉堆 (Binary Heap)

1. （类）完全二叉树结构，$depth(A) = O(\log|A|)$
2. priority(x) < priority(father(x))
3. 添加：上浮
4. 删除：下沉

可合并性质 (Mergeable)

1. 维护多个不相交集合最小值 / 最大值
   - 添加 / 删除元素
   - 合并两个集合
2. 二叉堆合并 $A, B \to C$：假设 $root(A) = root(C)$，将 B 与 $Left(A), Right(A)$ 中任一合并

左偏树 (Leftist Tree)

1. 合并时一直选择 $Left(A)$
2. 合并后保证左子堆的元素个数不大于右子堆的元素个数
3. 左子堆链深度为 $O(\log n) \Rightarrow$ 合并次数 $O(\log n)$
4. 添加、删除 合并
5. 并不满足 $depth(A) = O(\log |A|)$

Dispatching (APIO2012)

- 一棵有根树，每个点 i 有权值 c_i, l_i，
- 给定 C 选择一个点 a 以及其子树中的一个点集 S，使得

$$\sum_{i \in S} c_i < C$$

- 最大化 $l_a \times |S|$

线段树 (Segment Tree)

1. 维护区间信息
2. 树结构
   - $(0, n] \rightarrow (0, n/2] + (n/2, n]$
   - $(l, r) \rightarrow (l, mid] + (mid, r]$
3. 询问 $(a, b]$ in $(l, r]$ 满足前后缀则为 $O(\log n)$

树状数组 (Binary Index Tree)

1. $lowbit(l)$ 为 l 二进制表示中为 1 的最小位
2. 下标为 l 的节点维护 $(l-2^{lowbit(l)}, l]$ 的信息
3. 树结构表示：$father(l)=l-2^{lowbit(l)}$，多叉树
4. 实现：$2^{lowbit(l)}=l \wedge -l$
   - $-l(l>0)$ 补码（Complement）为 2^n-l+1

树状数组 (Binary Index Tree)

1. 询问 $(0, l]$ ：沿父链到根
2. 修改 x：迭代 $x+ = lowbit(x)$
3. 区间修改：差分

清华集训 2013 楼房重建 (Adapted)

- 序列 $a(a_1, \ldots, a_n)$
- $next(i)$ 为 i 后第一个比 a_i 大的元素的位置
- 单点修改，或求从 0 开始的 $next$ 链长度

树链剖分

1. 将树剖分成若干链（序列）来维护
2. 每一链沿着元素个数最多的子树下行
3. 任一链顶向上 $\Rightarrow$ 元素个数 $\times 2$
4. 任一点到树根最多经过 $O(\log n)$ 条链

Aragorn's Story (HDU3966)

1. 给一棵树以及各点权值
2. 若干操作
   - 将路径 (a, b) 上的所有点权加/减 K
   - 查询某个点的权值

函数式

1. 维护所有历史版本 T
2. 修改 $a_T \to$ 新建 a_{T+1}
3. 重定向版本 $T+1$ 的指针到 a_{T+1} 而非 a_T

函数式并查集

1. 启发式合并

DZY Loves Graph (UOJ Easy Round 1)

n 个点标号 $1 \sim n$

若干操作

- add a b 添加一条边 (a, b)，权值递增
- delete k 删去边权第 k 大的边
- return 撤销上一次操作，保证没有连续两次 return

函数式线段树

1. 修改的节点 $\Rightarrow$ 新建
2. 没有被修改的节点们（子树）$\Rightarrow$ 复制上一版本子树根节点的指针

区间 k 大

1. 给定一个序列 $a(a_1, \ldots, a_n)$
2. 每次询问区间 (l, r) 中第 k 大的值

清华集训 2014 mex

1. 给定一个序列 $a(a_1, \ldots, a_n)$
2. 每次询问 $(a_l, a_{l+1}, \ldots, a_r)$

函数式分块

1. 仅修改块 $B_T \to B_{T+1}$
2. 其他块复制指针

平衡树 (Balanced Binary Tree)

1. 二叉搜索树（Binary Search Tree)
2. 深度满足 $O(\log n)$
3. 支持添加 / 删除
4. 类似线段树，可维护区间信息

Treap

1. 每个元素随机 priority
2. priority 满足堆性质
3. 元素 a 深度 $\geq d$ 的概率为 2^{-d}

C++ STL

1. (multi) set <key [, cmp]>
2. (multi) map <key, value [, cmp]>

分治

- cdq： $F(\overline{ab}, \overline{ab}) = F(a, a) + G(a, b) + F(b, b)$
- 预处理结构
- 过分治点

双关键字最长上升子序列

NOIP2013 货车运输

1. 每次询问图上两点之间的路径经过的最大边权的最小值

k-Maximum Subsequence Sum (Adapted from CF 280 div1)

每次询问某个区间中选 k 个不相交连续子段的最大权值和

不带修改

分块

- 结构分块
- 询问分块
- 定期重建

COT2

1. 树上求两点之间不同的权值个数
2. OR (在线)

区间 k 大

1. 求区间中第 k 大的值，出现多次只算一次

Serega and Fun (cf 260 div1)

1. 给定一个序列 $a(a_1, \ldots, a_n)$
2. 每次将某个子序列 cyclic rotate
3. 询问 (l, r) 中等于 k 的元素个数