splay

陈哲章

北京大学

2025 年 7 月 16 日

1 splay
- 介绍
- 操作
- 复杂度分析

2 应用
- 普通平衡树

3 例题

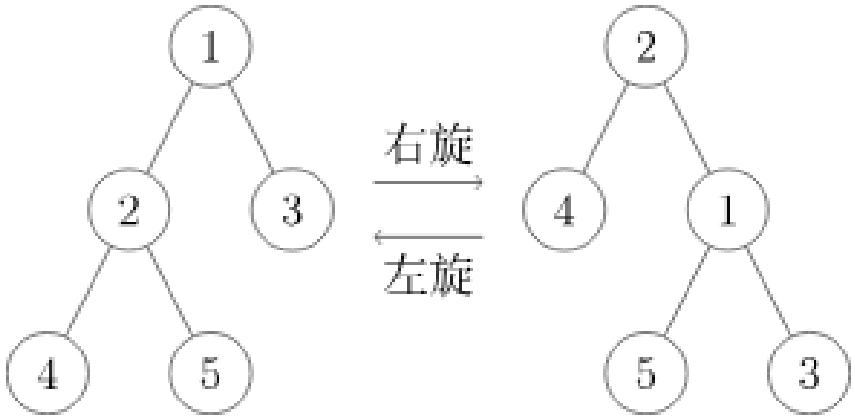
Splay 树由 Daniel Sleator 和 Robert Tarjan 于 1985 年发明。Splay 树是一种平衡二叉查找树，它利用伸展（splay）操作不断将某个节点旋转到根节点，使得他能够在均摊 $O(\log n)$ 得复杂度内完成平衡树得若干操作。

笔者习惯用结构体储存 splay 树的结点，以下是 splay 的必要信息：

- rt：根节点的编号。
- ->son[0/1]：左右结点的编号。
- ->fa：父亲节点的编号

其余信息根据不同题目的要求记录，如 ->val，->siz 之类的。

为了使 splay 保持平衡，需要进行旋转操作，将某个点向上移动一个位置，并且保持 splay 的二叉搜索树性质不变。
旋转分为左旋和右旋：



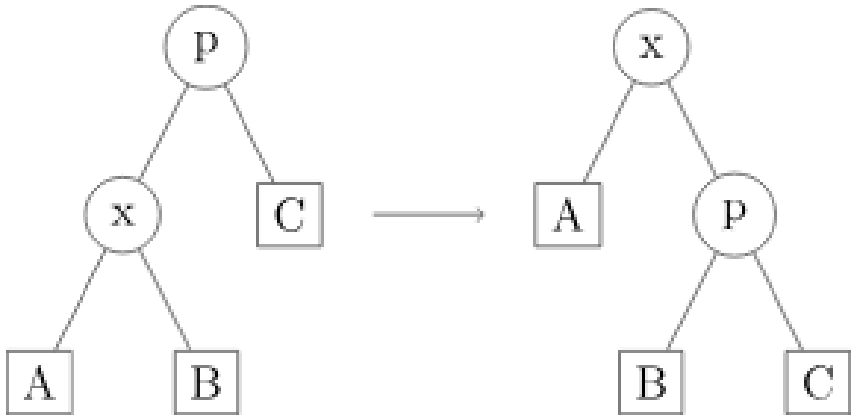
图：旋转操作

具体分析旋转的步骤（设当前结点为 x，父亲结点是 y，父亲的父亲为 z，且旋转是右旋）：

- 1. 自下而上更新信息：将 y 的左儿子标记为 x 的右儿子，x 的右儿子标记为 y，若 z 存在，将 z 对应的儿子标记成 x。(->fa 的信息也要一同更改)
- 2. 自下而上更新结点信息：先 y 再 x 最后到 z。

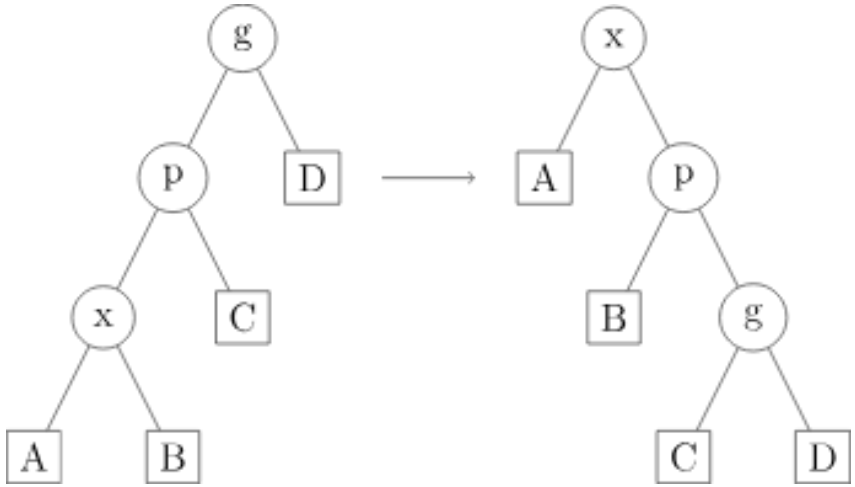
splay 的核心操作是伸展操作，要求每访问一个节点 x 都要将其旋转到根节点。
但是暴力地每次将 x 向上旋转一层是错误的！
我们需要用到双旋。
设当前结点是 x，父亲是 y，y 的父亲是 z。

1. zig：若 y 已经是根了（z 不存在），则直接旋转 x 就行了。



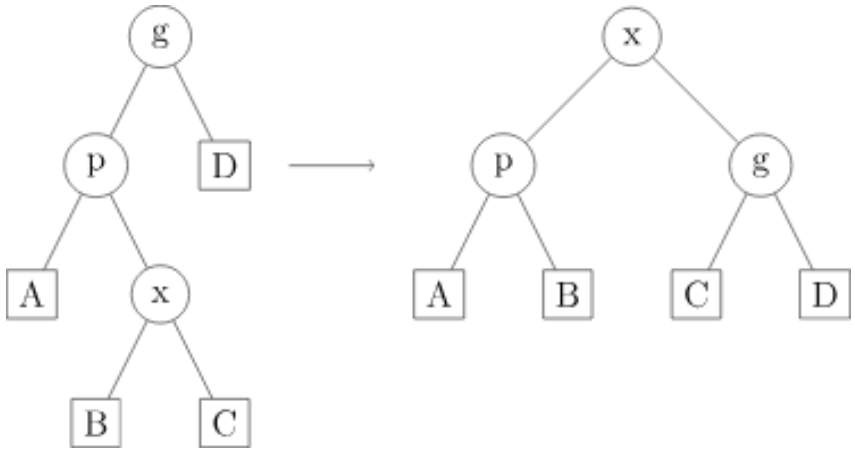
图：zig 操作

2. zig-zig：当 $x-y-z$ 这条链是直链（即是同侧儿子）的时候，我们先将 y 向上旋转一次，再将 x 旋转一次。



图：zig-zig

3. zig-zag：当 $x-y-z$ 这条链不是直链（即是异侧儿子）的时候，我们先将 x 向上旋转两次。



图：zig-zag

那么实现 splay 操作的时候，我们只需分别判断 x 是 y 的哪边儿子和 y 是 z 的哪边儿子然后分别采用不同的旋转策略就行了。
在平衡树操作中，每当我们要操作某个结点的时候，我们只需将其先 splay 到根节点然后直接改变它的信息即可。

先说结论，对大小为 n 的 Splay 树做 m 次伸展操作的复杂度是 $O((n+m)\log n)$ 的，均摊下来可以看作 $O(\log n)$ 的。

先说结论，对大小为 n 的 Splay 树做 m 次伸展操作的复杂度是 $O((n+m)\log n)$ 的，均摊下来可以看作 $O(\log n)$ 的。

我们定义单个结点的势能为 $w_x = \log siz_x$，即子树大小的对数，定义整棵树的势能为所有结点的势能之和，定义操作的成本 $c_i = t_i + w' - w$，其中 t_i 为该次操作的代价，w’ 为操作后的势能，w 为操作前的势能。

我们考虑分析不同操作的势能变化：

1. zig：势能发生变化的只有 x 点和 y 点，则

$$c_i = 1 + w'_x - w_x + w'_y - w_y = 1 + w'_y - w_x \le 1 + w'_x - w_x$$

2. zig-zig：同理，我们有

$$c_i = 2 + w'_x + w'_y + w'_z - w_x - w_y - w_z$$

由 $siz'_x = siz_x + siz'_z + 1$，由均值不等式得 $siz'^2_x \ge 4 siz_x siz'_z$，则 $w'_x \ge 2 + w_x + w'_z$。
则

$$c_i = 2w'_x - w_x - w'_z + w'_x + w'_y + w'_z - w_x - w_y - w_z = 2w'_x - 2w_x + w'_y - w_y$$

则

$$c_i \le 3(w'_x - w_x)$$

3. zig-zag：同理 $w'_x \geq 2 + w'_y + w'_z$，则有

$$c_i = 2w'_x - w'_y - w'_z + w'_x + w'_y + w'_z - w_x - w_y - w_z = 2w'_x - w_x - w_y$$

则

$$c_i \leq 2(w'_x - w_x) \leq 3(w'_x - w_x)$$

一次 Splay 下来总的变化量是 $c_i \leq 3(w'_x - w_x) = O(\log n)$ 的。
则单词操作的势能增加量不会超过 $O(\log n)$。
势能值的大小是 $O(n \log n)$ 的，则总的操作代价不会超过 $O(m \log n + n \log n)$。

Splay 可以实现普通平衡树的各种操作，大致和普通二叉搜索树一样，只多了一步 splay 到根。

- 1. 查找单个结点：无论按照值还是排名找，只需在二叉搜索树上找到那个结点然后将其 splay 到根就行了。
- 2. 插入单个结点：只需要在二叉搜索树上找到对应位置插入后将其 splay 到根。
- 3. 删除单个结点：先找到对应结点，splay 到根，删除它后我们得到了两棵 splay 树，只需将右子树的权值最小的结点 splay 到根，将左子树插入到右子树的左儿子。

- 4. 查排名：额外记录每个结点的子树大小，然后找单个结点的同时计算排名，记得查询完后 splay 到根。
- 5. 查前驱/后继：直接在对应二叉搜索树上查询即可，记得最后 splay 到根。

模板：P3369 【模板】普通平衡树;P6136 【模板】普通平衡树（数据加强版）

splay 还可以做区间操作

- 1. 区间加减/查询以操作区间是 $[l, r]$ 为例，我们先把 $l-1$ splay 到根，然后把 $r+1$ splay 到 $l-1$ 的右儿子，此时 $l-1$ 的左子树就表示 $[l, r]$ 这个区间。
- 2. 打标记区间操作很重要的一个方法就是打标记，但是 splay 的旋转操作会影响标记的使用，所以在 splay 之前要把从根到当前点路径上的标记从上往下下放。

模板：P3391 【模板】文艺平衡树

- P3850 [TJOI2007] 书架
- P4008 [NOI2003] 文本编辑器
- P2042 [NOI2005] 维护数列
- P5338 [TJOI2019] 甲苯先生的滚榜
- P1110 [ZJOI2007] 报表统计