Splay树

前置知识
讲解044-前缀树　整个专题的要求

有序表专题安排
专题1：AVL树，讲解148
专题2：跳表，讲解149
专题3：替罪羊树，讲解150
专题4：笛卡尔树、Treap树，讲解151
专题5：FHQ Treap树，讲解152
专题6：Splay树，讲解153，本节

大厂笔试、算法竞赛掌握以上有序表结构足够，其他有序表结构不再讲述，面试遇到只是聊，可以自行学习
算法竞赛的同学，有序表必带模版：替罪羊树、Treep树、FHQ Treap树、Splay树
Splay树是实现Link-Cut-Tree的关键，这个结构的讲述，会在【挺难】阶段的课程里安排

左程云

Splay树

本节课的前置知识
讲解110 - 线段树，了解懒更新机制
讲解148 - AVL树，了解左旋操作、右旋操作
讲解152 - FHQ Treap，进行比较学习

本节课讲述
Splay树的提根操作
Splay树实现有序表及其实战题目，题目1、题目2
势能分析法计算Splay树的时间复杂度
Splay树的注意点
Splay树解决区间移动、范围修改的问题，题目3、题目4、题目5

Splay树的发明者是我的偶像，Robert Tarjan，因为诸多贡献获得了1986年的图灵奖

Splay树

Splay树的提根操作
提根操作是指，节点a通过上升的方式，变成节点b的儿子，或者整棵树的头节点
提根的过程不能破坏搜索二叉树的性质

节点a在上升的过程中，每一步都讨论，当前点、父亲点、爷爷点之间的关系
根据不同的情况，进行不同的调整，具体分为三种情况
1，升一步的调整(zig)，上升一步就到达最终位置了，那么当前点向上一次即可
2，一字型的调整(zig-zig)，爷爷、父亲、当前点是一字型关系，父节点先向上，当前点再向上
3，之字型的调整(zig-zag)，爷爷、父亲、当前点是之字型关系，当前点向上两次

底层节点上升的过程，根据上述策略进行调整
不管什么样的长链，底层节点提根到头节点后，长链高度几乎会减少一半

课上重点图解提根的过程，提根操作的代码 + 长链高度变化的实验，ShowDetail文件展示

Splay树

Splay树实现有序表
1，查询过程
2，插入过程
3，删除过程

本节课的实现一律约定，Splay树维护的搜索二叉树为，左 < 头 <= 右

重要方法，int find(int rank) ： 整棵树上找到中序排名为rank的节点，返回节点编号

加入过程、删除过程都有提根操作，同时要保证每种查询功能都有提根操作！这很重要！

课上重点图解，Splay树实现有序表的过程和代码

Splay树

题目1
Splay树的实现，不用词频压缩
实现一种结构，支持如下操作，要求单次调用的时间复杂度O(log n)
1，增加x，重复加入算多个词频
2，删除x，如果有多个，只删掉一个
3，查询x的排名，x的排名为，比x小的数的个数+1
4，查询数据中排名为x的数
5，查询x的前驱，x的前驱为，小于x的数中最大的数，不存在返回整数最小值
6，查询x的后继，x的后继为，大于x的数中最小的数，不存在返回整数最大值
所有操作的次数 <= 10^5
-10^7 <= x <= +10^7
测试链接 : https://www.luogu.com.cn/problem/P3369

本节课的实现一律约定，Splay树维护的搜索二叉树为，左 < 头 <= 右

Splay树

势能分析法，摊还分析的一种，此外还有聚合分析法、核算法，有兴趣的同学看《算法导论》第17章
势能借用了物理学的概念，可以理解为预付代价，积攒势能的释放，可以用于支付未来操作的代价

$$\hat{c}_i = c_i + \Phi(D_i) - \Phi(D_{i-1})$$

$$\sum_{i=1}^{n} \hat{c}_i = \sum_{i=1}^{n} (c_i + \Phi(D_i) - \Phi(D_{i-1})) = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$$

1. $\hat{c}_i$：第 i 次操作的均摊代价。
2. c_i：第 i 次操作的实际代价。
3. $\Phi(D_i)$：操作完成后，系统状态 D_i 的势能。
4. $\Phi(D_{i-1})$：操作开始前，系统状态 D_{i-1} 的势能。
5. $\Phi(D_n)$ 和 $\Phi(D_0)$：分别是第 n 次操作结束后的势能和初始状态的势能。

先确定整个过程的复杂度预期上界，Φ函数的选取不能超过该上界，然后分析复杂度能否收敛于该上界
Φ函数的选取还需要考虑，系统特征、计算便利性等因素
课上重点图解，k位二进制计算器，使用势能分析法时，推荐工具视角！

左程云

Splay树

不同场景 w, s, r, Φ 的定义不同，以下都是对 Splay 树的分析场景来说的

$w(u) = 1$

$s(u) =$ 以 u 为头的子树节点数

$r(u) = \log(s(u))$

$\Phi(u) = \sum_{i \in \text{tree}(u)} r(i)$

1. $w(u)$：节点 u 的权值，每个节点权值为 1。
2. $s(u)$：整棵子树的权值，就是子树上所有节点的数量。
3. $r(u)$：单个节点的势能函数，取子树大小的对数 $\log(s(u))$。
4. $\Phi(u)$：整棵子树的势能函数，等于子树中所有节点势能的总和。

课上重点图解，Splay树的势能分析

Splay树的r、Φ函数，网上戏称为黑魔法，需要灵感和观察，简要说明为什么如此定义，推荐工具视角！

Splay树进行n次操作的总时间复杂度O(n * log n)，单次操作的时间复杂度可以看作O(log n)

左程云

Splay树

Splay树的注意点

1，初始化时，可以使用二分的方式递归建出完美的平衡树，其实没有必要
　　势能分析的证明过程说明，初始不管平衡还是不平衡，都不会影响单次均摊的时间复杂度
　　哪怕出现极度不平衡的情况，随着操作的进行，长链会迅速变短，树会快速变成近乎平衡的状况
2，Splay树单次操作的时间复杂度O(log n)，但是常数项时间较大
3，不管是插入、删除、查询，要保证都有提根操作，让长链得到缩减机会
4，Splay树的实现，慎用递归，虽然长链会得到缩减，但仍可能出现长链，让递归爆栈
5，大多数时候，Splay树可以被FHQ Treap替代，本节课所有题目，都可以使用FHQ Treap解决
6，可持久化Splay树不需要掌握，因为空间耗损较大，掌握可持久化FHQ Treap即可
7，Splay树具有独特的提根操作，可以帮助实现Link-Cut-Tree，【挺难】阶段会安排讲述

Splay树

题目2
郁闷的出纳员
最低薪水为limit，一旦员工薪水低于limit，员工会离职，实现如下四种操作
I x ： 新来员工初始薪水是x，如果x低于limit，该员工不会入职当然也不算离职
A x ： 所有员工的薪水都加上x
S x ： 所有员工的薪水都减去x，一旦有员工低于limit那么就会离职
F x ： 查询第x多的工资，如果x大于当前员工数量，打印-1
所有操作完成后，打印有多少员工在操作期间离开了公司
测试链接 ： https://www.luogu.com.cn/problem/P1486

Splay树

题目3
文艺平衡树，Splay实现范围翻转，java版本
长度为n的序列，下标从1开始，一开始序列为1，2，...，n
接下来会有k个操作，每个操作给定l，r，表示从l到r范围上的所有数字翻转
做完k次操作后，从左到右打印所有数字
1 <= n，k <= 10^5
测试链接 ： https://www.luogu.com.cn/problem/P3391

Splay树

题目4
书架
给定一个长度为n的排列，由数字1、2、3...n组成，实现如下五种操作
Top s　　：数字s移动到最左边
Bottom s　：数字s移动到最右边
Insert s t ：数字s位置假设为rank，现在移动到rank+t位置
Ask s　　：查询数字s左边有多少数字
Query s　：查询从左往右第s位的数字
所有操作保证都是合法的
测试链接 ： https://www.luogu.com.cn/problem/P2596

Splay树

题目5
维护数列
初始时给定一个数列，实现如下六种操作
INSERT posi tot ... : 在第posi个数字之后，插入长度为tot的数组，由...代表
DELETE posi tot : 从第posi个数字开始，删除长度为tot的部分
MAKE-SAME posi tot c : 从第posi个数字开始，长度为tot的部分，值都设置成c
REVERSE posi tot : 从第posi个数字开始，翻转长度为tot的部分
GET-SUM posi tot : 从第posi个数字开始，查询长度为tot的部分的累加和
MAX-SUM : 查询整个数列中，非空子数组的最大累加和
任何时刻输入保证至少有一个数字在数列中，并且所有操作都合法
插入数字总数很多，但是任何时刻数列中最多有5 * 10^5个数，使用总空间要和该数量有关
测试链接 : https://www.luogu.com.cn/problem/P2042

区间移动 + 区间更新 + 区间翻转 + Splay树维护懒更新信息

左程云