

# Splay

如何用  $\text{Splay}$  维护二叉查找树

## 简介

$\text{Splay}$  是一种二叉查找树，它通过不断将某个结点旋转到根结点，使得整棵树仍然满足二叉查找树的性质，并且保持平衡而不至于退化为链，它由 Daniel Sleator 和 Robert Tarjan 发明。

## 结构

### 二叉查找树的性质

首先肯定是一棵二叉树！

能够在这棵树上查找某个值的性质：左子树任意结点的值  $<$  根结点的值  $<$  右子树任意结点的值。

### 结点维护信息

$\text{rt}$	$\text{tot}$	$\text{fa}[i]$	$\text{ch}[i][0/1]$	$\text{val}[i]$	$\text{cnt}[i]$	$\text{sz}[i]$
根结点编号	结点个数	父亲	左右儿子编号	结点权值	权值出现次数	子树大小

## 操作

### 基本操作

- $\text{maintain}(x)$  在改变结点位置后，将结点  $x$  的  $\text{size}$  更新
- $\text{get}(x)$  判断结点  $x$  是父亲结点的左儿子还是右儿子。
- $\text{clear}(x)$  销毁结点  $x$

```
void maintain(int x) { sz[x] = sz[ch[x][0]] + sz[ch[x][1]] + cnt[x]; }
bool get(int x) { return x == ch[fa[x]][1]; }
void clear(int x) { ch[x][0] = ch[x][1] = fa[x] = val[x] = sz[x] = cnt[x] = 0; }
```

### 旋转操作

为了使  $\text{Splay}$  保持平衡而进行旋转操作，旋转的本质是将某个结点上移一个位置。

#### 旋转需要保证

- 整棵  $\text{Splay}$  的中序遍历不变（不能破坏二叉查找树的性质）。
- 受影响的结点维护的信息依然正确有效。
- $\text{root}$  必须指向旋转后的根结点。

在  $\text{Splay}$  中旋转分为两种：左旋和右旋。



From:  
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:lgwza:splay&rev=1598083393](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:lgwza:splay&rev=1598083393)



Last update: **2020/08/22 16:03**