**RANK-BALANCED TREE 作业三**

计科三班 1140310323 蒲毅

## 对于平衡二叉树研究的局限性

之前的关于平衡二叉树的研究都只在于平衡二叉树的实现以及优化和分析方面，每一种平衡树都有自己的定义标准，不能很好的统一定义平衡二叉树这一体系，本文新提出了一种从未被系统化的rank理论，并用这一理论体系重新定义了常见的平衡二叉树，而且还提供了一种基于该理论的复杂度均摊分析的方法，用来比较这些平衡树的性能，在此基础上还提出了一种新型的平衡二叉树，在性能上优于红黑树以及AVL。

## 1.rank的理论和定义

之前的关于平衡二叉树的研究都只在于平衡二叉树的实现以及优化和分析方面，而本文对于平衡二叉树引入了一个没有人提出过的分类方法，通过rank来对平衡树进行分类，而我们如下对rank 进行定义：

(1)：对所有节点赋予一个非负的正整数 rank，记为r(x)，对于外部点（不含有信息的节点）rank 为 -1.

(2)：平衡二叉树的rank定义为这棵平衡二叉树的根的rank，即rank(root)。

(3)：如果节点x的父亲节点是p(x)，定义节点x的rank差为r(p(x)) – r(x)。

(4)：非根节点x称为a节点如果x的rank差为a

(5)：定义x为（a，b）节点如果x的孩子的rank差分别为a，b。

## 2.用rank来定义常用的平衡二叉树

有了上面的定义，可以对平衡二叉树进行一个大体的分类。

文章定义了两种分类方法：

(1)：所有的rank差为[1，c]，其中c >= 2为常数。

(2)：所有的rank差为[0，c]，不含有连续的超过 c’ 的路径的节点rank差为0 。

其中c>=0, c’ >=0。

有了以上的分类方法，常见的平衡二叉树可以如下定义，注意没有列举对称的情况：

(1)AVL ：每个节点为（1，2） 节点或（1，1）节点。

(2)2-3树：每个节点是（1，1）节点或（0，1）节点，且0节点的父亲不为0节点。

(3)red-black tree：每个节点都为0节点或1节点，且0节点的父亲不为0节点。

(4)右倾2-3树：每个节点为（1，1）或（0，1），0节点的父亲不为0节点，0节点不为左孩子。

(5)左倾2-3树：每个节点为（1，1）或（0，1），0节点的父亲不为0节点，0节点不为右孩子。

(6)右倾红黑树：每个节点为（1，1）或（0，1）或（0，0），0节点的父亲不为0节点，且0节点不为（0，1）节点的左孩子。

(7)左倾红黑树：每个节点为（1，1）或（0，1）或（0，0），0节点的父亲不为0节点，且0节点不为（0，1）节点的右孩子。

以上可以得出结论，我们可以将我们常用的平衡二叉树都用一种新的定义方式来得到一个统一的定义方法。使得平衡二叉树的理论更加的系统化。

## 3.根据新的理论定义新的平衡二叉树

通过新的rank理论体系来定义了常见的平衡二叉树，发现平衡树的本质的区别就是允许节点的类型差异，所以本文直接从这种新理论的本质出发定义了一种全新的平衡二叉树WAVL

WAVL：所有节点为1节点或2节点，所有的叶子结点为0节点。

相比于AVL的定义较为宽松，所以称其为 weak AVL。

**WAVL 调整平衡：**

定义操作 promotion ：将节点的rank+1

定义操作 demotion ：将节点的rank-1

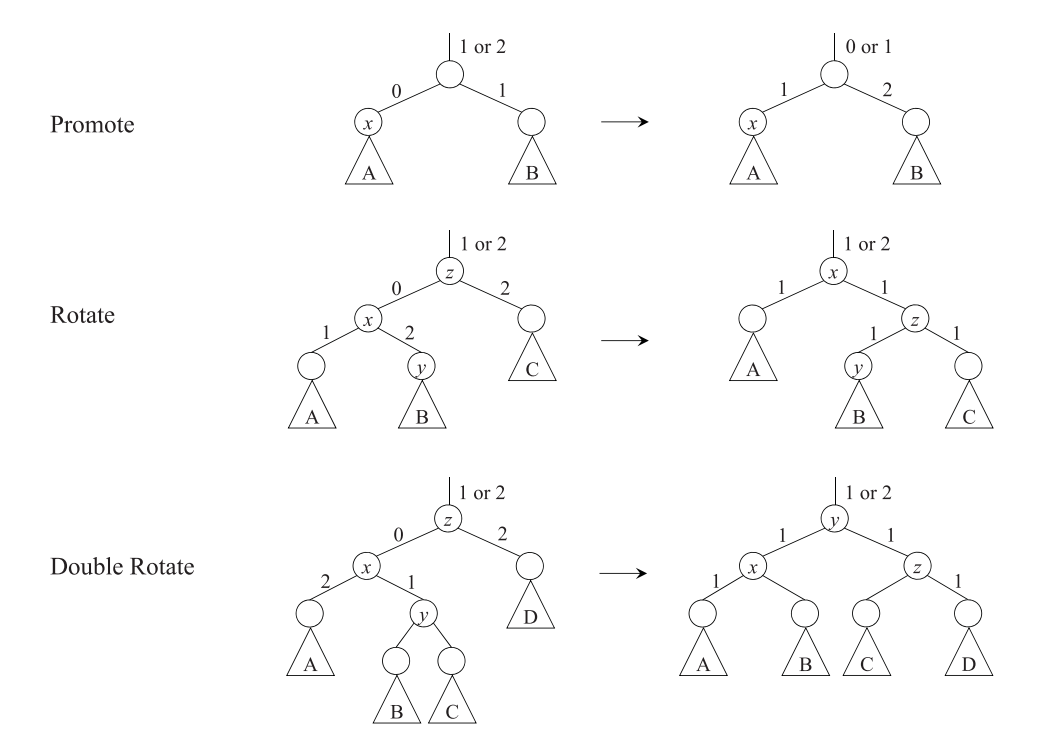
插入操作需要调整平衡的情况:

插入节点后使得其父亲节点从（1，1）节点变成了（0，1）节点，所以我们执行以下步骤：

While p(x) = null and p(x) is （0, 1）

Promote p(x). Replace x by p(x).

在此操作之后根据得到的树的情况进行旋转操作：



删除操作需要调整平衡的情况：

While x is a 3节点 and y is a 2节点 or （2，2）节点

Demote:

{

If y is a 2节点 :

demote p(x);

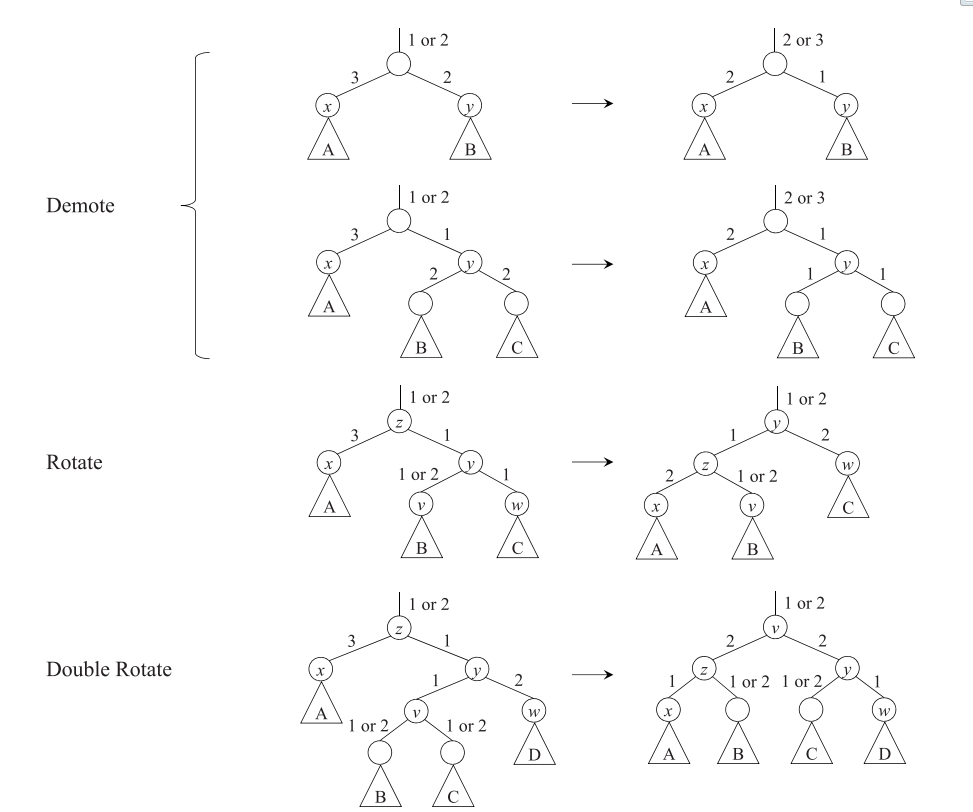
Otherwise:

demote both y and p(x).

let x = p(x), and let y be the sibling of x.

}

然后根据对应下图的不同形式调整树重新倒平衡



## 4.rank理论的优势以及WAVL的性能优势

通过rank理论我们重新定义了常见的平衡树，使得平衡二叉树这一领域有了全新的认识和分析的角度。也便于继续对平衡二叉树的理论和实用性研究。

WAVL在没有删除操作的情况下，插入节点如果使树失去平衡，跟新的路径最大长度为logφn ≈ 1.44log2n(即树的最大高度为logφn ≈ 1.44log2n)；如果有删除操作存在，那么跟新的路径最大长度为2log2n 。并且在发现失衡之后，可以在做多两次旋转操作将树重新调整到平衡。所以WAVL拥有AVL以及RBH共同的优点。