替罪羊树

替罪羊树

- 引入替罪羊树的目的
- 替罪羊树的定义
- 替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- 总结

引入替罪羊树的目的

相比普通的二叉树，那些具有平衡功能的二叉树的节点要花费更多的内存开销，如AVL树存储平衡因子，红黑树存储颜色。

替罪羊树（除根节点外）无需存储额外的信息，相比别的平衡树减少了内存开销，并且时间复杂度相同。

替罪羊树

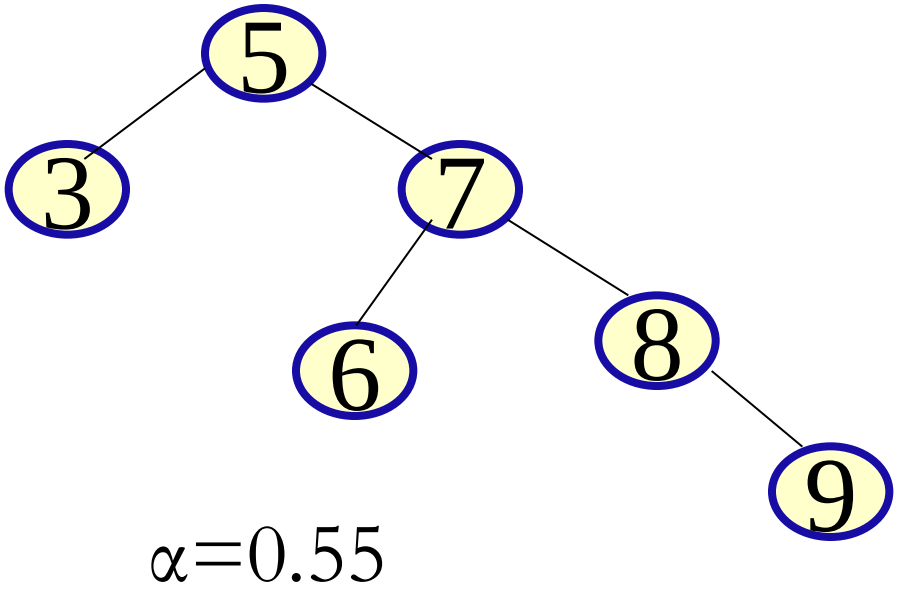
- ✓引入替罪羊树的目的
- ➢替罪羊树的定义
- ◘替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- ◘总结

两个概念

对于二叉排序树的根结点

1）若满足$h<=h_{\alpha}(h_{\alpha}=\lfloor \log_{1/\alpha} n \rfloor)$则称它为α高度平衡（AVL 红黑树）

2）若满足$h<=h_{\alpha}+1$，则称它为宽松α高度平衡



α=0.55

$h_{\alpha}=\lfloor \log_{1/0.55} 6 \rfloor =2$

h=3

不满足$h<=h_{\alpha}$

不是一棵0.55高度平衡树

满足$h<=h_{\alpha}+1$

是一棵宽松0.55高度平衡树

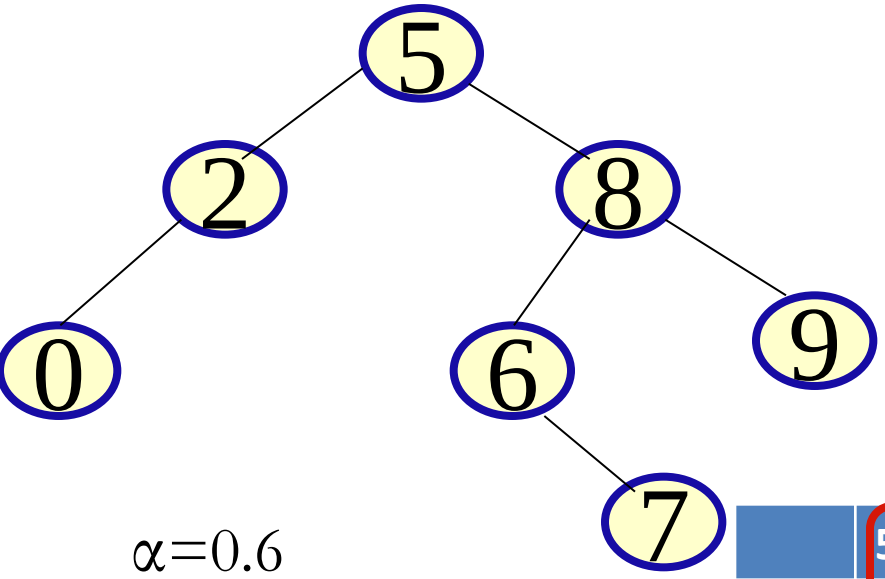
当n一定时

α值越小，二叉树越稠密，插入效率越低，查询效率越高

α值越大，二叉树越稀疏，插入效率越高，查询效率越低

α权重平衡

若二叉树排序树的每个结点都满足$n_{左}<=\alpha*n$ 并且$n_{右}<=\alpha*n$
则称它为α权重平衡



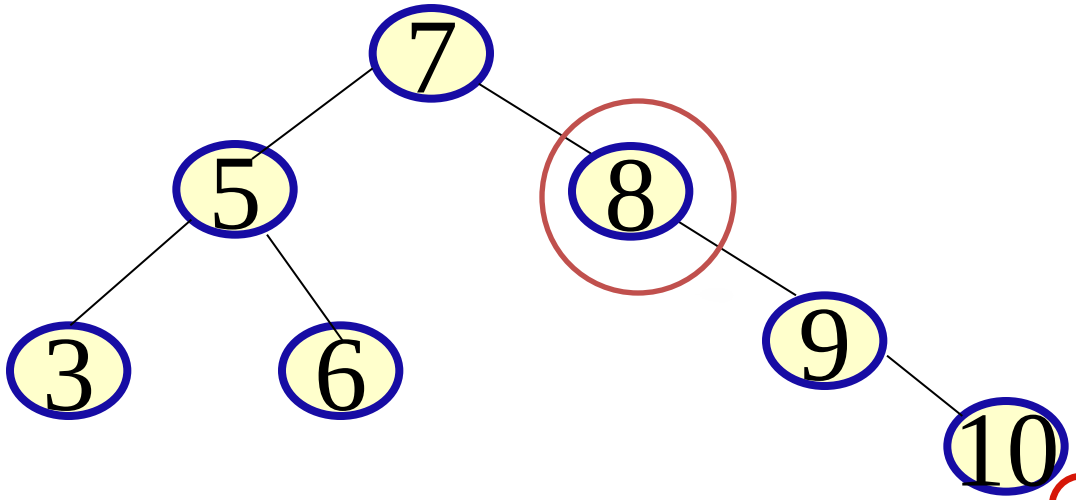
α=0.6

所以它是一棵0.6权重平衡排序树

	5	2	8	0	6	9	7
α	0.6	0.6	0.6	0.6	0.6	0.6	0.6
n	7	2	4	1	2	1	1
α*n	4.2	1.2	2.4	0.6	1.2	0.6	0.6
$n_{左}$	2	1	2	0	0	0	0
$n_{右}$	4	0	1	0	1	0	0
	平衡	平衡	平衡	平衡	平衡	平衡	平衡

α权重平衡

若二叉树排序树的每个结点都满足$n_{左}<=\alpha*n$ 并且$n_{右}<=\alpha*n$
则称它为α权重平衡(AVL)



不是一棵0.6权重平衡二叉树

α=0.6

	7	5	8	3	6	9	10
α	0.6	0.6	0.6	0.6	0.6	0.6	0.6
n	7	2	3	1	1	2	1
α*n	4.2	1.2	1.8	0.6	0.6	1.2	0.6
$n_{左}$	3	1	0	0	0	0	0
$n_{右}$	3	1	2	0	0	1	0
	平衡	平衡	不平衡	平衡	平衡	平衡	平衡

特殊情况 当二叉树为0.5权重平衡时

即$n_{左}<=0.5*n$ 并且$n_{右}<=0.5*n$

则几乎为完全二叉树

证： $n-n_{右}-1= n_{左}$

$0.5n-1 =< n-n_{右}-1 = n_{左} =< 0.5n$

则$n_{左}\approx 0.5n$ $n_{右}\approx 0.5n$

树t几乎为完全二叉树

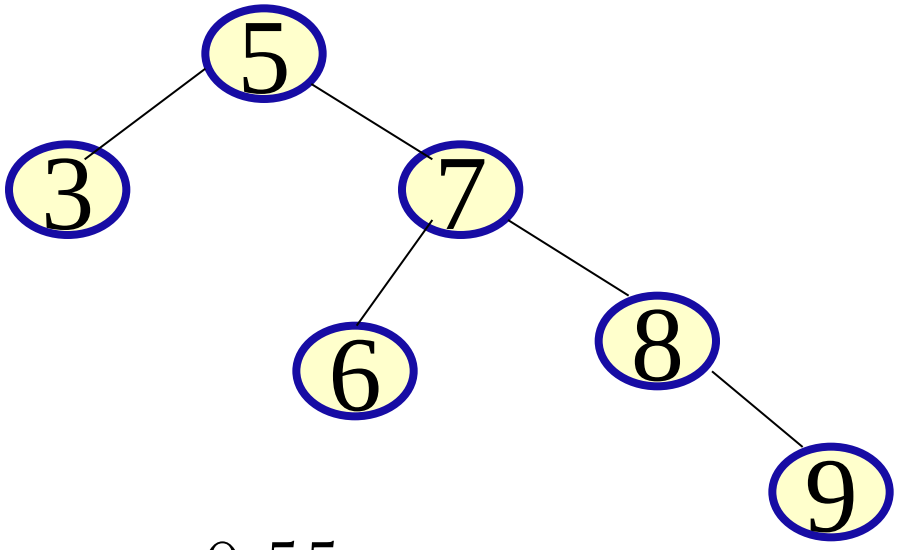
定理：如果一棵二叉树α权重平衡，那它一定α高度平衡
反之不一定成立

替罪羊树(scapegoat tree)的定义:

高度平衡与重量平衡的结合

- 是一种二叉排序树
- 根节点存储了树的结点总个数n和上次重建后的结点个数$n_{上次}$
- 总能保证宽松的α高度平衡

  即$h<=h_\alpha+1$($h_\alpha=\lfloor \log_{1/\alpha} n \rfloor$)



$\alpha=0.55$

$h_\alpha=\lfloor \log_{1/0.55} 6 \rfloor =2$

$h=3$

满足$h<=h\,\alpha+1$

是一棵宽松0.55高度平衡树

是一棵替罪羊树

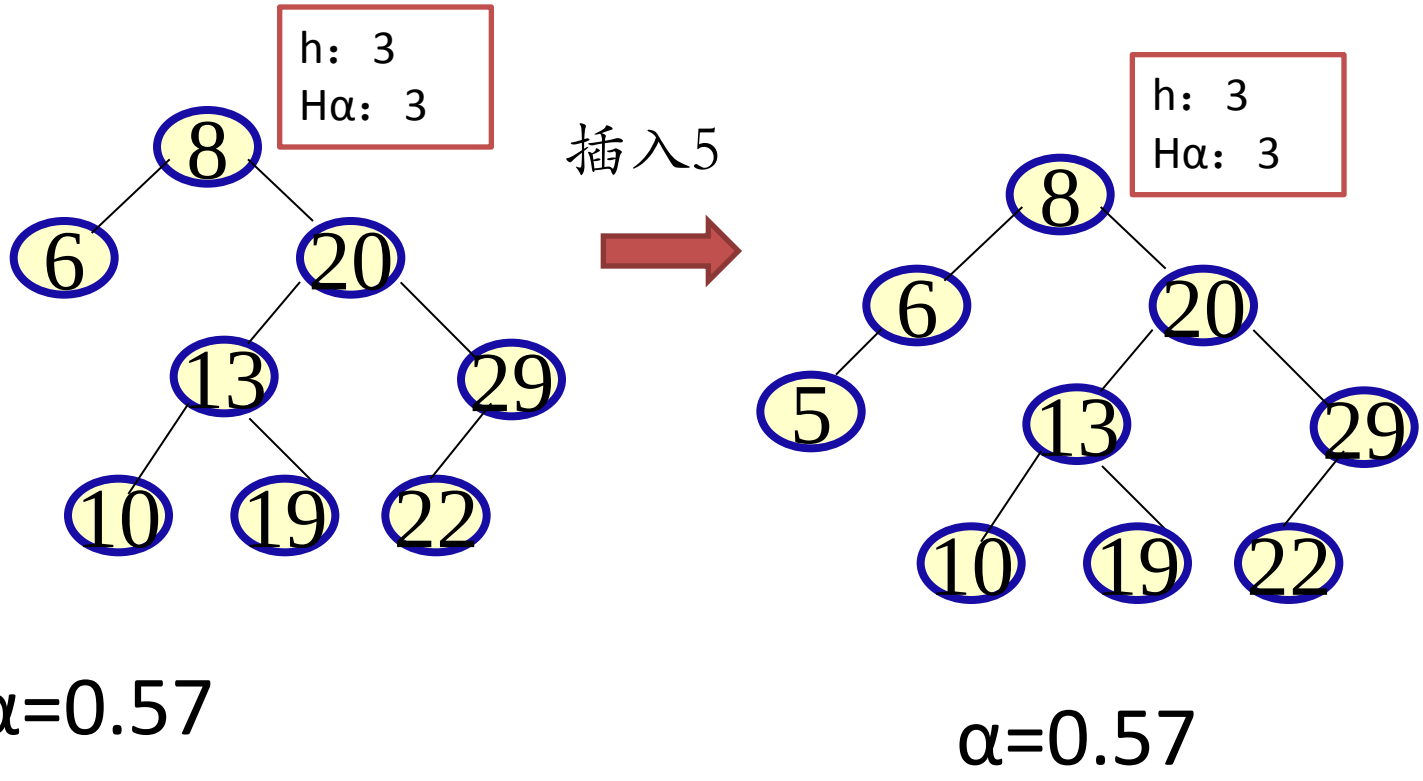
替罪羊树

- ✓引入替罪羊树的目的
- ✓替罪羊树的定义
- ➢替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- ◘总结

α高度平衡:h<=hα(hα= $\lfloor \log_{1/\alpha} n \rfloor$)
宽松的α高度衡: h<= hα+1

替罪羊树的插入

1） 当插入新节点后仍保持α的高度平衡，
则和普通的二叉排序树的插入方法一致。



α=0.57

α=0.57

替罪羊树的插入

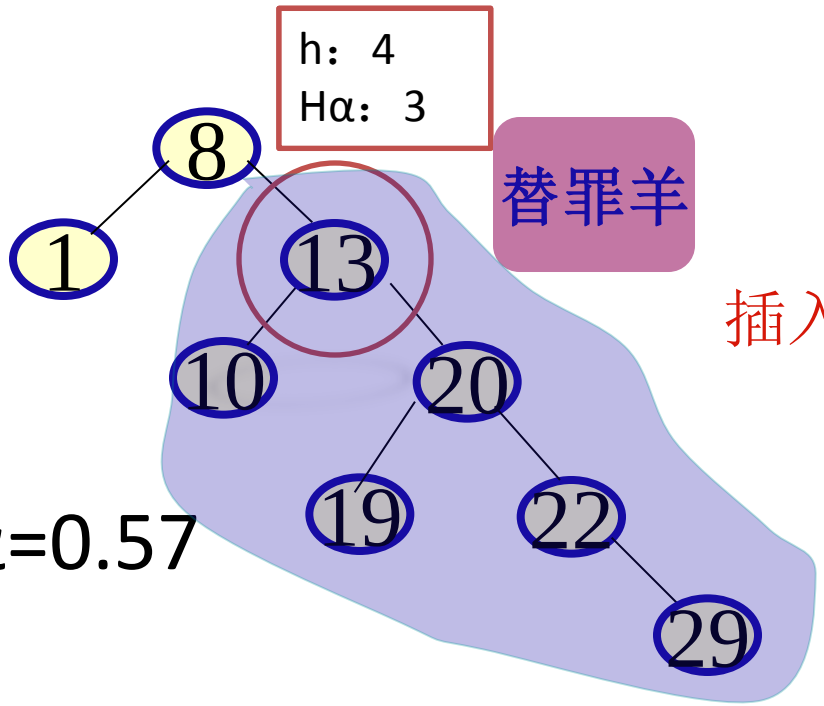
α高度平衡:h<=hα(hα= $\lfloor \log_{1/\alpha} n \rfloor$)
宽松的α高度衡:　h<= hα+1
α权重平衡:$n_{左}$<=α*n　&& $n_{右}$<=α*n

2）当插入新节点后打破了α的高度平衡，则要寻找替罪羊，把以替罪羊为根节点的子树重建成一个新的0.5权重平衡（即完全二叉树）的二叉排序树

如何寻找替罪羊?

沿着新插入节点的双亲向上找，第一个不满足α权重平衡的结点即为替罪羊。

替罪羊树的插入

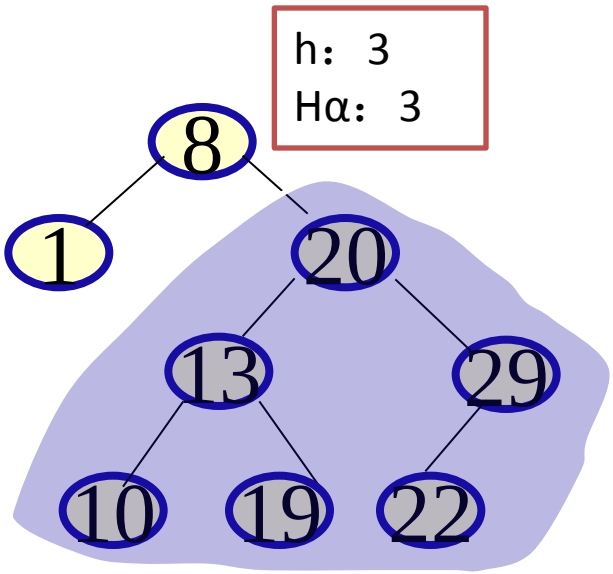


α=0.57

插入29

α高度平衡:h<=hα(hα= $\lfloor \log_{1/\alpha} n \rfloor$)
宽松的α高度衡: h<= hα+1
α权重平衡:$n_{左}$<=α*n && $n_{右}$<=α*n

平均： O(log(n))



$n_{上次}$=8

	22	20	13
α	0.57	0.57	0.57
n	2	4	6
α*n	1.14	2.28	3.42
$n_{左}$	0	1	1
$n_{右}$	1	2	4
	平衡	平衡	不平衡

替罪羊树

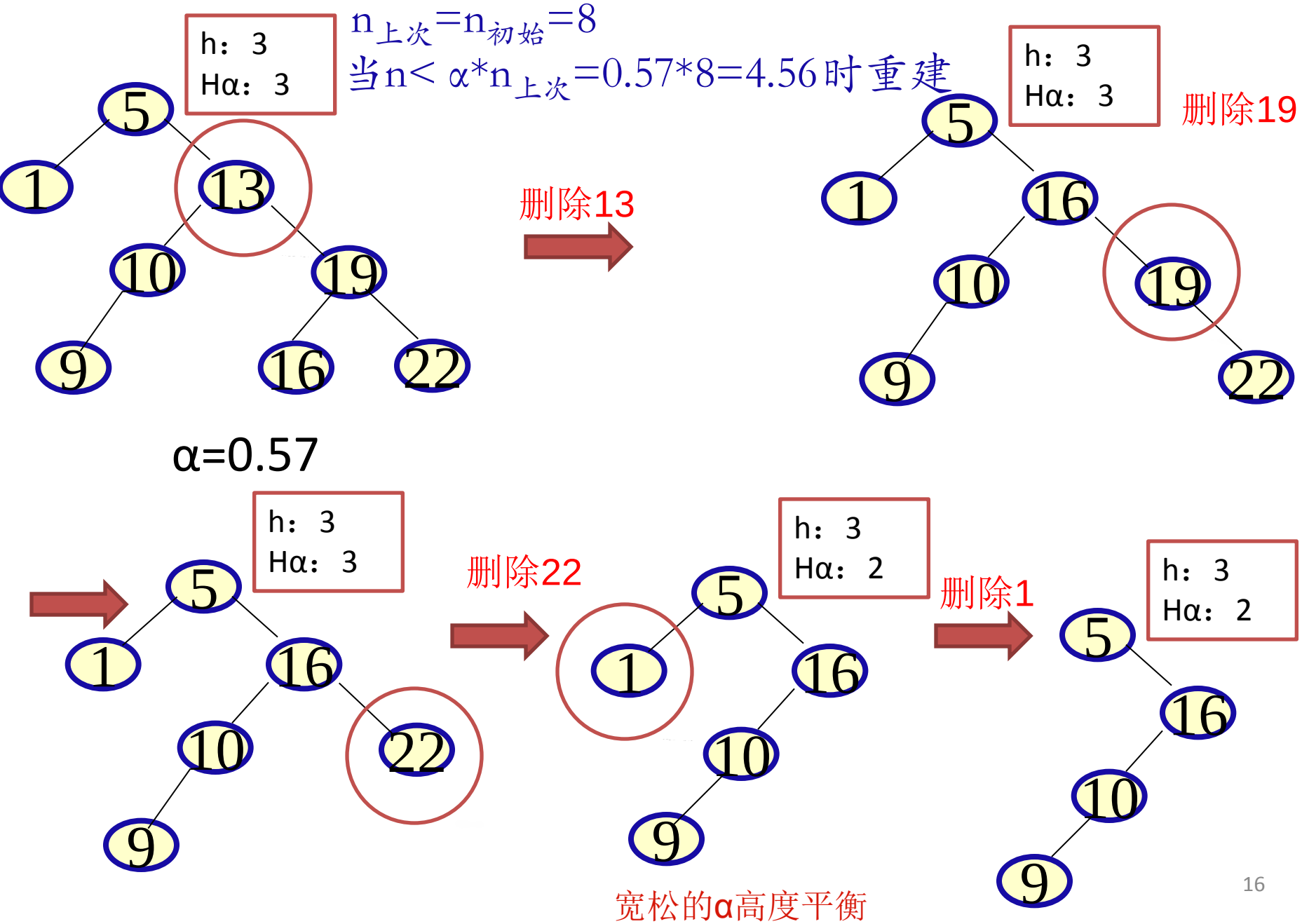
- ✓引入替罪羊树的目的
- ✓替罪羊树的定义
- ➢替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- ◘总结

替罪羊树的删除

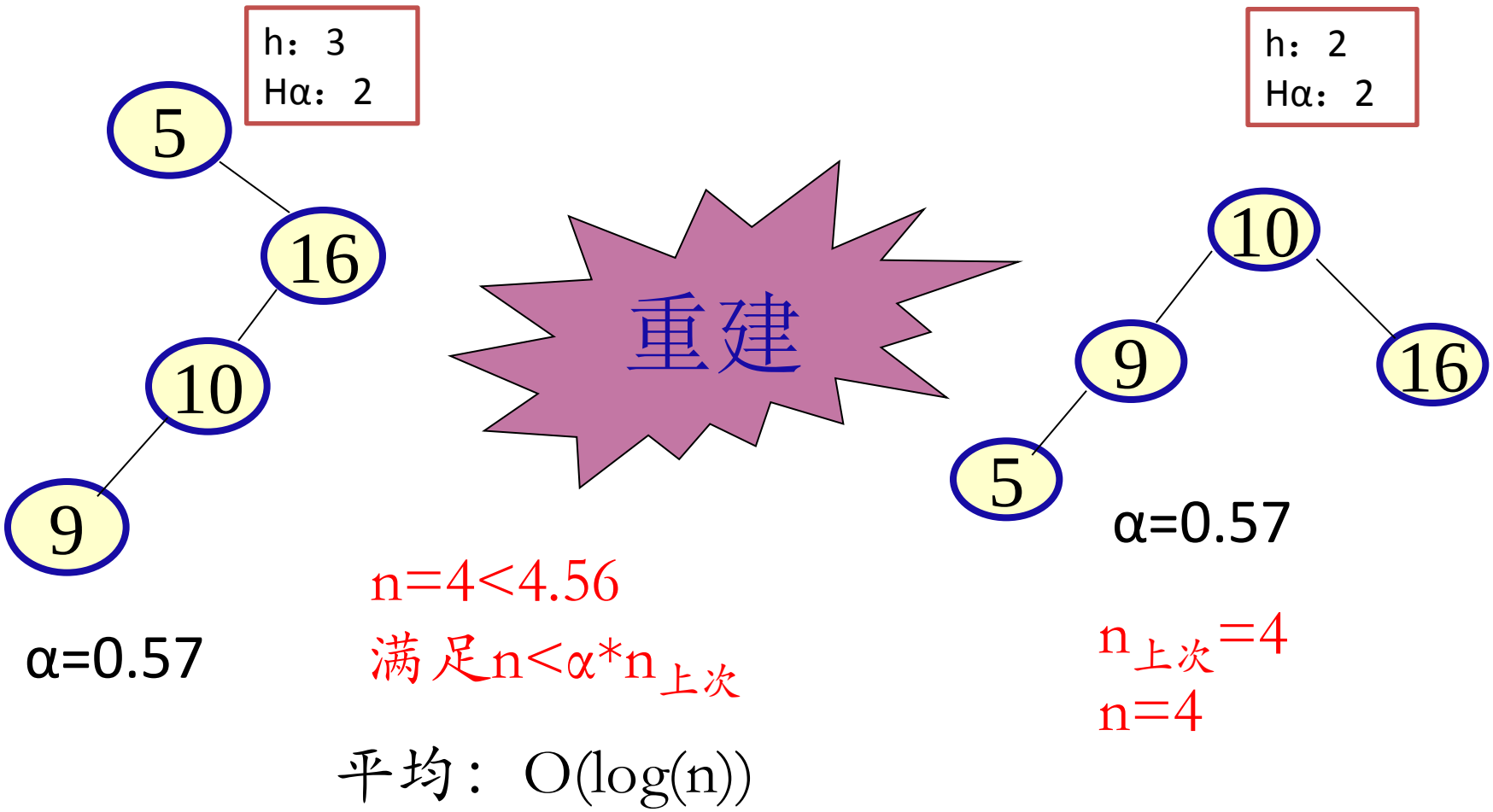
一般情况下，替罪羊树的删除和普通的二叉排序树的删除一样。

不过，当每次删除完成后，若$n<\alpha*n_{上次}$，则整个树将重建成0.5权重平衡（即完全二叉树）的二叉排序树。
（$n_{上次}$为二叉树上次重建后的结点个数）

替罪羊树的删除操作


h：3
Hα：3
$n_{上次}=n_{初始}=8$
当$n<\alpha*n_{上次}=0.57*8=4.56$时重建
删除13
h：3
Hα：3
删除19
α=0.57
h：3
Hα：3
删除22
h：3
Hα：2
删除1
h：3
Hα：2
宽松的α高度平衡

替罪羊树的删除操作



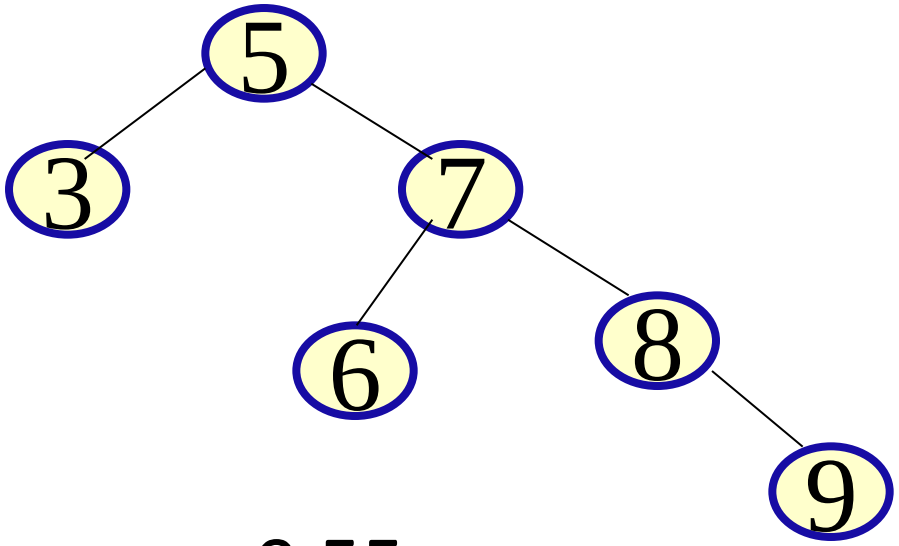
无论插入还是删除替罪羊树总能保证宽松的α高度平衡

替罪羊树

- ✓认识几个基本概念
- ✓引入替罪羊树的目的
- ✓替罪羊树的定义
- ➢替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- □总结

替罪羊树的查找

替罪羊树的查找操作和普通的二叉排序树的操作一样



α=0.55

最坏情况：O(log(n))

替罪羊树

- ✓认识几个基本概念
- ✓引入替罪羊树的目的
- ✓替罪羊树的定义
- ✓替罪羊树的基本操作
  - 替罪羊树的插入
  - 替罪羊树的删除
  - 替罪羊树的查找
- ➢总结

总结

1.替罪羊树不同于别的大部分平衡树，它的节点（除根节点）不用存储额外的信息（例如颜色，平衡度）。（根节点存储了整个树的节点个数n和上次重建后的结点个数$n_{上次}$）.从而节约了存储空间。

2.替罪羊树不用旋转来达到平衡，操作简单。但是替罪羊树需要重建。

Thanks