

目录

CONTENTS

01

【平衡二叉树的定义】

02

【如何构造平衡二叉树】

03

【平衡二叉树的查找性能分析】

01

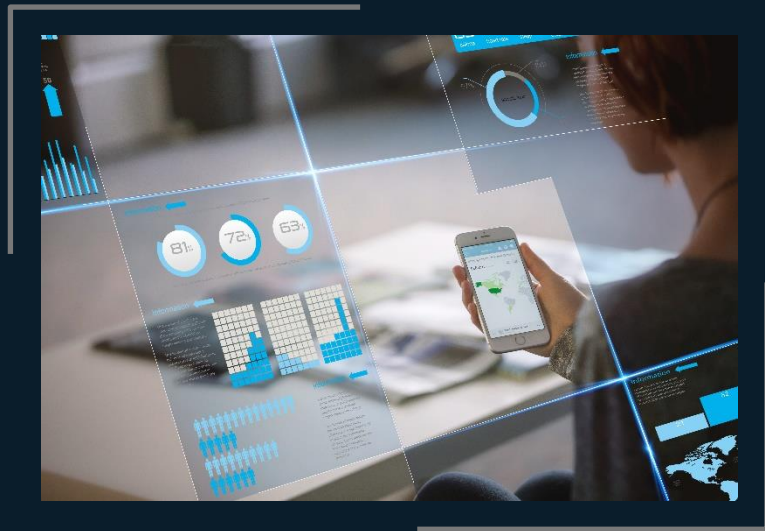
平衡二叉树的定义



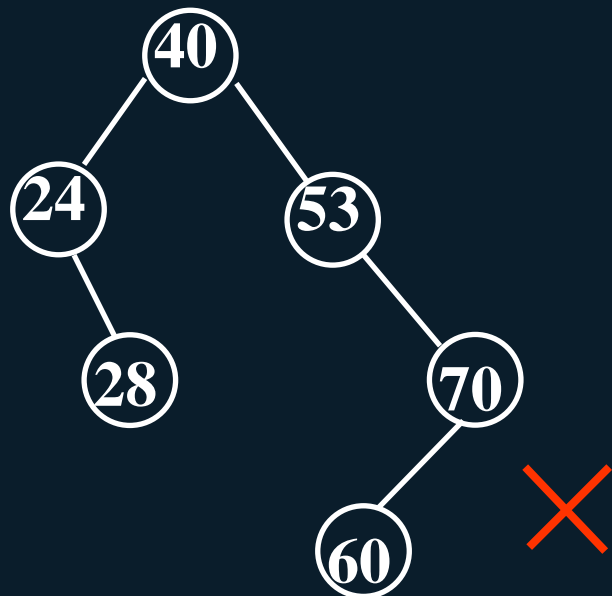
平衡二叉树的定义

一棵AVL树或者是空树，或者是具有下列性质的二叉树：

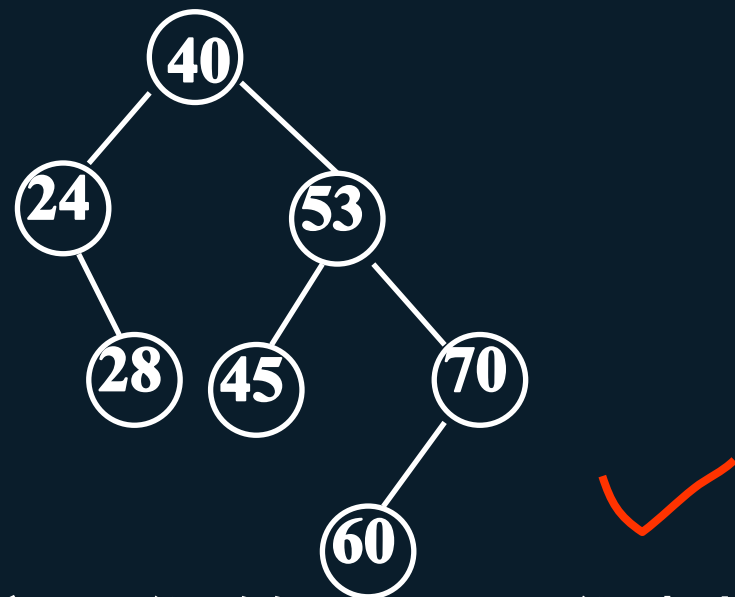
- (1) 它的左子树和右子树都是AVL树；
- (2) 二叉树的左子树和右子树的高度之差的绝对值不超过1。



二叉平衡树 (AVL) 高度平衡的二叉搜索树



高度不平衡的二叉平衡树



高度平衡的二叉平衡树

结点的平衡因子 *balance* (balance factor)

- 每个结点附加一个数字，给出该结点左子树的高度减去右子树的高度所得的高度差。这个数字即为结点的平衡因子 *balance*。
- 根据AVL树的定义，任一结点的平衡因子只能取 -1，0和 1。
- 如果一个结点的平衡因子的绝对值大于1，则这棵二叉搜索树就失去了平衡，不再是AVL树。



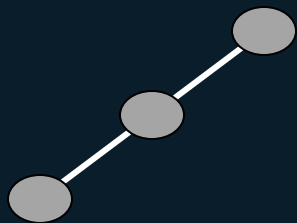
02

如何构造平衡二叉树

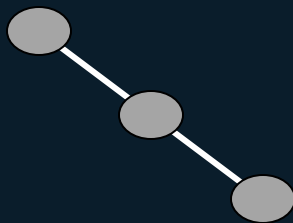
构造二叉平衡树

- 如果在一棵平衡的二叉搜索树中插入一个新结点，造成了不平衡。此时必须调整树的结构，使之平衡化。
- 平衡化旋转有两类：
 - 单旋转 (左旋和右旋)
 - 双旋转 (左平衡和右平衡)
- 每插入一个新结点时，AVL树中相关结点的平衡状态会发生改变。因此，在插入一个新结点后，需要：**从插入位置沿通向根的路径回溯，检查各结点的平衡因子(左、右子树的高度差)。**
- 如果在某一结点发现高度不平衡，停止回溯。

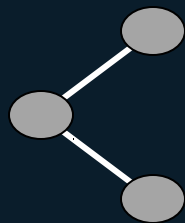
- 从发生不平衡的结点起，沿刚才回溯的路径取直接下两层的结点。
- 如果这三个结点处于一条直线上，则采用单旋转进行平衡化。单旋转可按其方向分为左单旋转和右单旋转，其中一个是另一个的镜像，其方向与不平衡的形状相关。
- 如果这三个结点处于一条折线上，则采用双旋转进行平衡化。双旋转分为先左后右和先右后左两类。



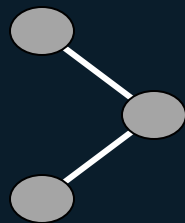
右单旋转



左单旋转

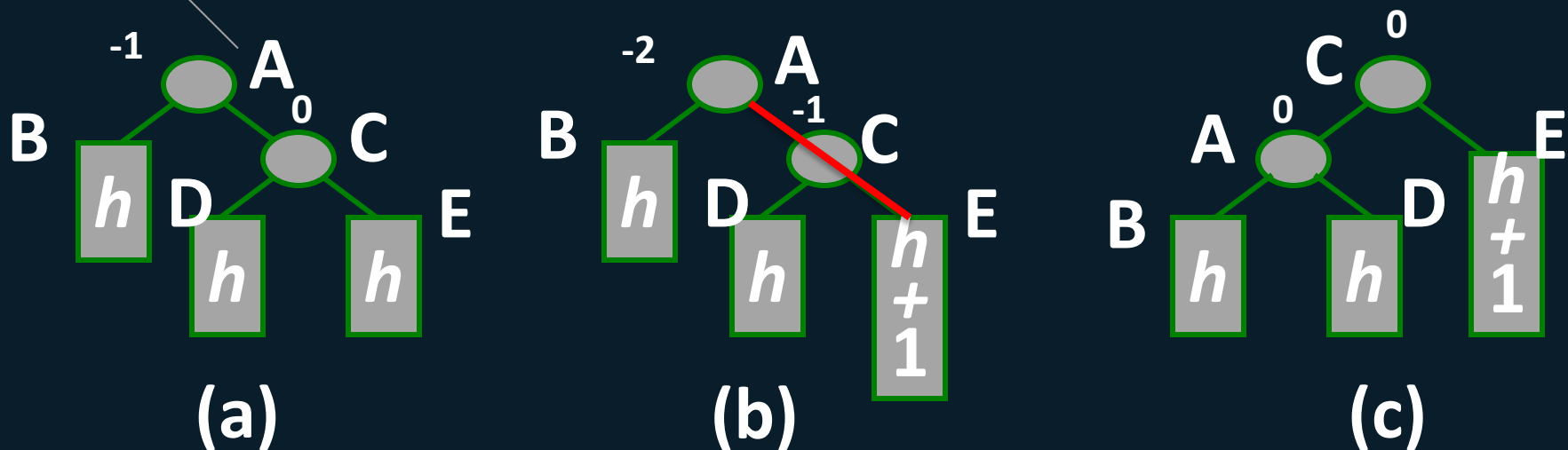


左右双旋转



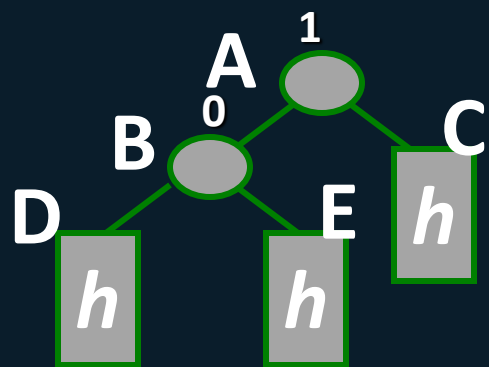
右左双旋转

1.左单旋转 (RotateLeft)

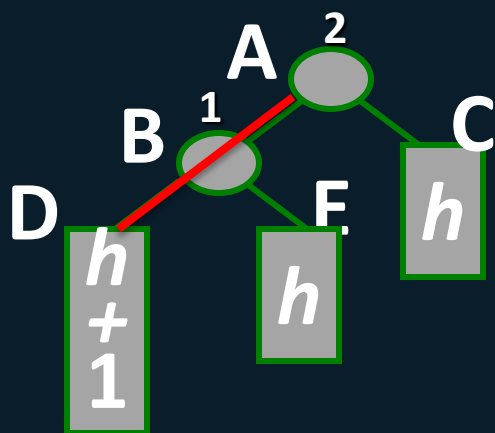


- 如果在子树E中插入一个新结点，该子树高度增1导致结点A的平衡因子变成 -2 ，出现不平衡。
- 沿插入路径检查三个结点A、C和E。它们处于一条方向为“\”的直线上，需要做左单旋转。
- 以结点C为旋转轴，让结点A反时针旋转。

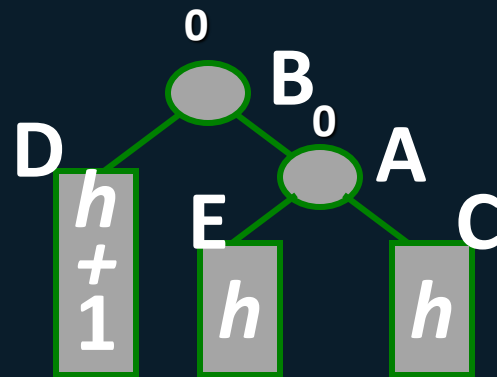
2.右单旋转 (RotateRight)



(a)



(b)



(c)

- 在左子树D上插入新结点使其高度增1，导致结点A的平衡因子增到+2，造成了不平衡。
- 为使树恢复平衡，从A沿插入路径连续取3个结点A、B和D，它们处于一条方向为“/”的直线上，需要做右单旋转。
- 以结点B为旋转轴，将结点A顺时针旋转。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/486142201115010100>