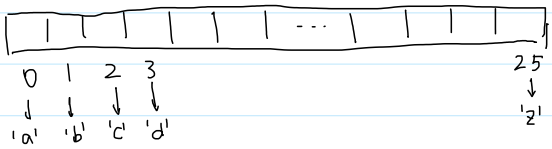
哈希表

2022年3月24日 10:03

Q. 统计每个小写字母出现的次数？

('a', 1)　('b', 3)　('c', 0) ...　键值对

key ← 　↳ value



键：① 范围不大

② 键可以转换成数组的索引

key - 'a'

int hash(K key)

↓ 索引　↓ 键

打散键值对（让键值对尽量平均分布）

V get(key) → O(1)

void put(key, value) → O(1)

void remove(key) → O(1)

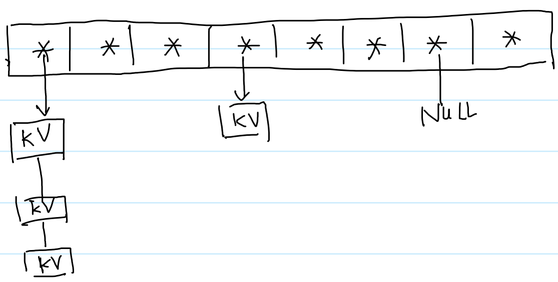
哈希表的两个问题：

① 哈希函数

让键值对尽量平均分布

② 解决冲突

拉链法



哈希表的基本操作

V get(K key)：

int idx = hash(key).

遍历链表

key存在，返回key对应的value

key不存在，返回特殊值.

V put(K key, V value):

int idx = hash(key);

遍历链表：

key存在，更新key对应的value，并返回原来的value

key不存在，添加键值对，返回特殊值.

V remove(K key)

int idx = hash(key);

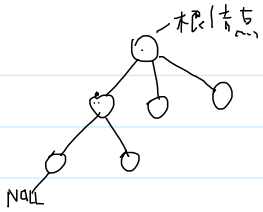
遍历链表：

key存在：删除键值对，并返回删除对应的值

key不存在：返回特殊值.

二叉搜索树 (BST)

2022年3月24日 11:32



Tree

族谱：Family Tree

父结点：

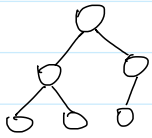
孩子结点：

兄弟结点：

根结点

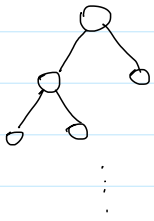
叶子结点：① 没有孩子的结点 ② NULL结点

二叉树：



每个结点最多有两个孩子

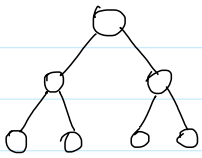
二叉树的性质：



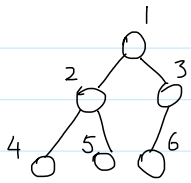
层	结点
1	$\leq 2^0$
2	$\leq 2^1$
3	$\leq 2^2$
⋮	
n	$\leq 2^{n-1}$

如果一棵树有n层，那它最多有 2^n-1 个结点。

特殊的二叉性：



满二叉树

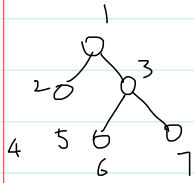


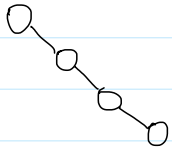
完全二叉树

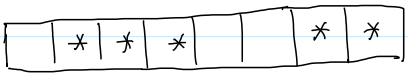
leftchild(i) = 2i

rightchild(i) = 2i + 1

parent(i) = i/2 (NULL → 0)

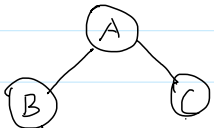






二叉树可以用数组的方式表示。

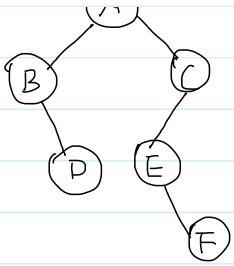
#1. 二叉树的遍历



深度优先遍历

先序遍历：DLR

ABDCEF



先序遍历：DLR
A B D C E F

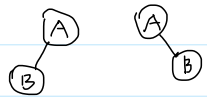
中序遍历：LDR
B D A E F C

后序遍历：LRD
D B F E C A

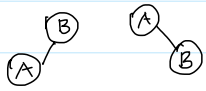
广度优先遍历（层级遍历）
A B C D E F

#2. 二叉树的建树.

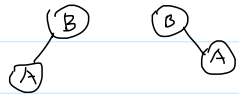
先序：A B



中序：A B

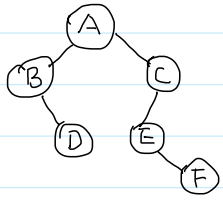


后序：A B



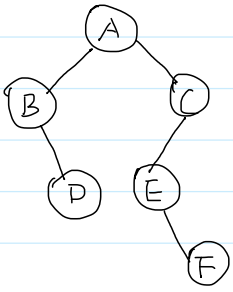
先序：A B D C E F

中序：B D A E F C

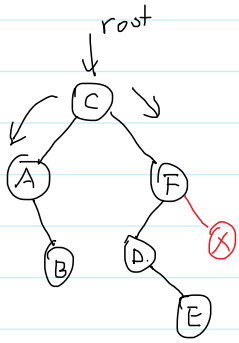


#3. 二叉查找树（Binary Search Tree）

1. 二叉树
2. 左子树所有结点的key值都小于根结点的key值.
   右子树所有结点的key值都大于根结点的key值.
   并且左、右子树都是二叉查找树.



（不是）



（A B C D E F）

查找：O(h) h为树的高度

插入：O(h)

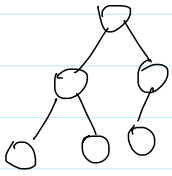
删除：O(h)

Q1：如何判定一棵二叉树是不是BST？（中序）

Q2：二叉查找树的好处.

Q3：如果一棵BST有n个结点，那么它最小的是多少？
完全二叉树.

一棵高度为h的完全二叉树，它结点数目的范围是多少？



$\leq 2^0$

$\leq 2^1$

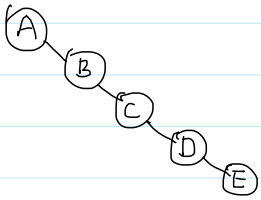
$\leq 2^2$ $h=2$

$2^h \leq n \leq 2^{h+1}-1$

$\Rightarrow 2^h \leq n < 2^{h+1}$

$\Rightarrow \log_2 n - k\ h \leq \log_2 n$ $\lfloor \log_2 n \rfloor$（向下取整）

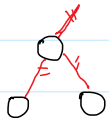
普通的二叉查找树的高度不一定是O(logn)这个级别！



平衡二叉查找树：

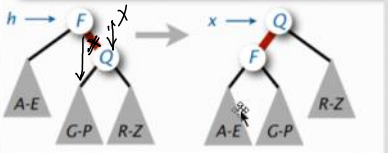
AVL树：对任意一个结点，它的左子树的高度和右子树的高度相差不超过1

红黑树：保证树的高度是O(logn)级别）

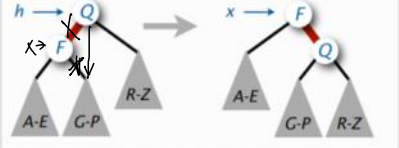


旋转

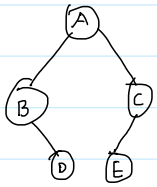
```
private Node rotateLeft(Node h)
{
   Node x = h.right;
   h.right = x.left;
   x.left = h;
   x.color = x.left.color;
   x.left.color = RED;
   return x;
}
```




```
private Node rotateRight(Node h)
{
   Node x = h.left;
   h.left = x.right;
   x.right = h;
   x.color = x.right.color;
   x.right.color = RED;
   return x;
}
```



用队列实现广度优先遍历



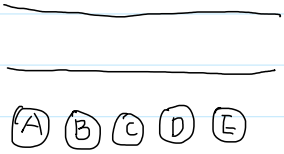
① 将根结点入队列

② 判断队列是否为空

非空：

出队列，获取结点

判断该结点是否有左孩子



有，入队到

判断该结点是否有右孩子

有，入队到

回到②

空，结束循环.