

A

1. 在一棵高度为 2 的 5 阶 B 树中, 所含关键字的个数最少是\_\_\_\_\_。

A . 5                      B . 7                      C . 8                      D . 14

C

2. 一棵高度为 3 的 3 阶 B 树 T 中, 第 2 层 (根为第 1 层) 有 4 个关键字, T 中结点个数最多为\_\_\_\_\_。

B

3. 7 个关键字的 4 阶 B 树可能的结构个数是\_\_\_\_\_。

A . 8                      B . 9                      C . 10                      D . 11

C

4. 一棵红黑树的根结点的阶 (根结点到外部结点路径上黑色边的数目) 为 2, 则其结点数目 (不包括外部结点) 最多为\_\_\_\_\_。

A . 13                      B . 14                      C . 15                      D . 16

A

5. 红黑树中根结点到某个外部结点路径长度为  $L$ , 其中红边数量为  $R$ , 则根结点到另一外部结点路径长度最小为\_\_\_\_\_。

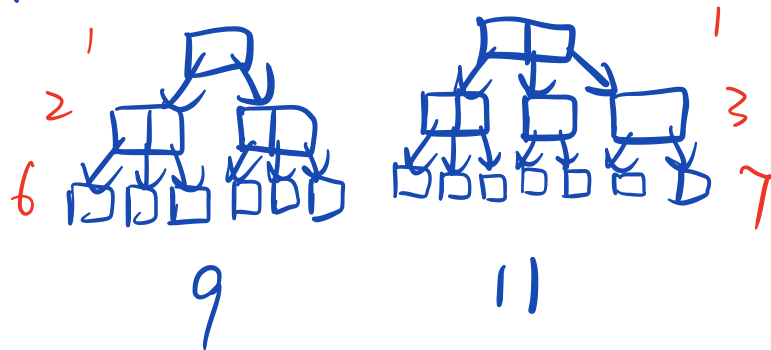
A .  $L-R$                       B .  $2(L-R)$                       C .  $L$                       D .  $L+R$

6. 若矩阵  $A_{m \times n}$  中的某一元素  $A[i][j]$  是第  $i$  行中的最小值, 同时又是第  $j$  列中的最大值, 则称此元素为该矩阵的一个鞍点。假设以二维数组存放矩阵, 试使用 C 或 C++ 语言编写一个算法, 确定鞍点在数组中的位置 (若鞍点存在时), 并分析该算法的时间复杂度。

7. 如何使用图表示好友关系? 好友关系是等价关系吗?

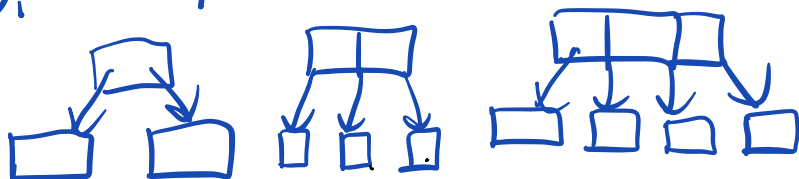
1. 5阶: 2~4个key 

2. 3阶: 1~2个key



4个key 结点个数越多越合适

3. 4阶: 1~3个key



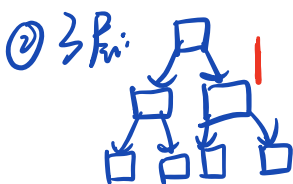
6个key:

5个key

4个key

① 2层: 3+3

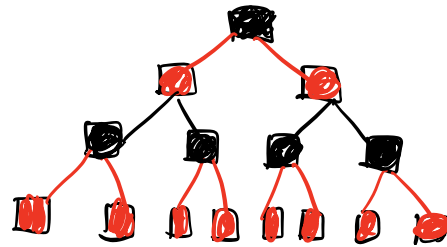
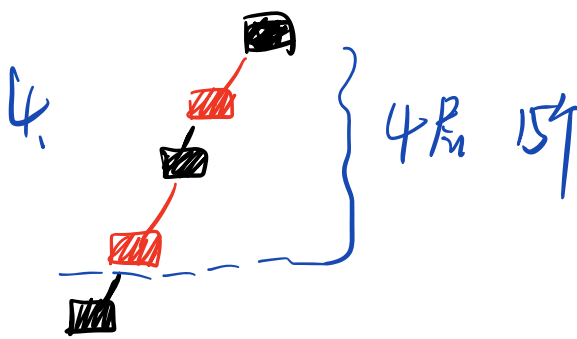
② 2层:



5个key在3层

1+1+3

1+2+2



第一步：创建 $a[m][n]$ 和 $b[m][n]$ ， $b[m][n]$ 与 $a[m][n]$ 对应。

第二步：输入矩阵A，并将数组b的元素全部初始化为0；

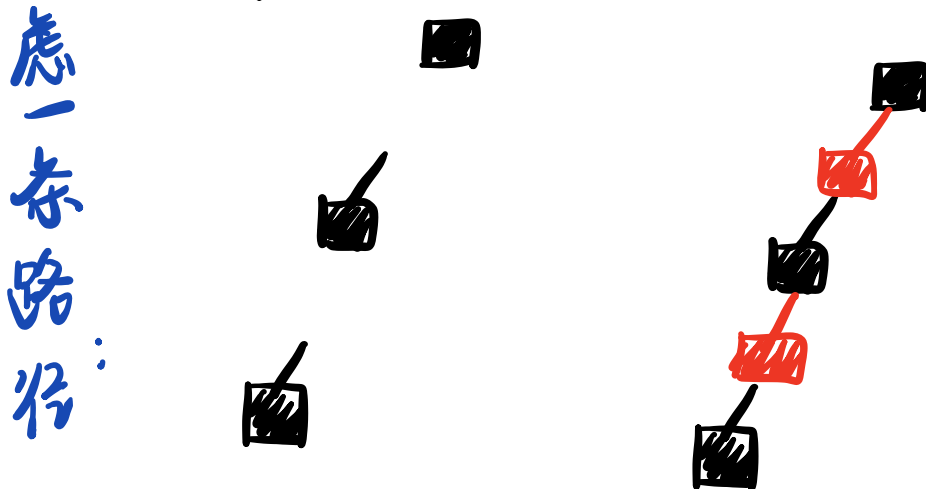
第三步：判断矩阵A中的每个元素是不是所在行中最小的数（由于可能有多个数相同且为最小，所以应该先确定最小的数的值，然后再判断该行中各元素值是否等于该值），是则让其对应的数组b中的元素值加1。

第四步：和第三步一样的操作，判断矩阵A中的元素是否为所在列中最大的数，是则让其对应的数组b中的元素值加1。

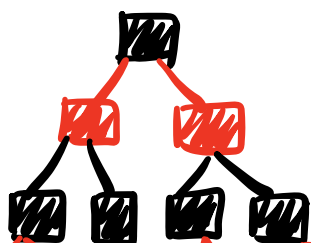
第五步：这时，数组b中的元素全部由0、1、2三个数字构成，如果某元素值为2，则说明其对应的数组a中的元素既是所在行中最小的元素，也是所在列中最大的元素，即鞍点。

考虑①先画黑点：


②再插红点



考虑全树：



} 前4层共15个节点

 → 这层不用算