

张铭

2006-12-29

Array[0][0][0]:1024000(字节)

存储形式：行优先

求: Array[5][7][9] (起止位置)

考点：理解行优先和列优先元素的存储顺序

解答: 依题意数组的存储一次为 $(0, 0, 0) \rightarrow (0, 0, 1) \cdots \rightarrow (0, 0, 49)$
 $\rightarrow (0, 1, 0) \cdots \rightarrow (0, 2, 0) \rightarrow (1, 0, 0) \cdots \rightarrow (19, 29, 49)$

开始位置: $1024000 + (5 \times 30 \times 50 + 7 \times 50 + 9) \times 8 = 1086872$

注意: 起止位置: 1086872--1086879

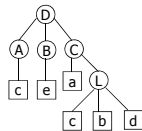
1) 长度: 3 (元素个数)

深度: 3 (括号的层数)

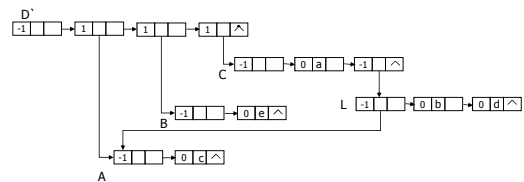
2) 这道题的含义就是重用子表A, 注意单个元素是不能被重用的, 必须定义成一个子表, 才能复用。

$$D^-(A(c), B(e), C(a, L(A, b, d)))$$

3) 参考答案:



(4) 参考答案



数据结构：链表（按块的起始位置来组织块的顺序）

初始状态: 整个空间表就是一块,

分配: 找满足分配要求且最小的空间块; 然后分配要求的空间, 剩余空间仍留在空利用空间表中;

回收：注意相邻空闲结点的合并，减少外部碎片。

对于下面的一组值{cup,cop,copy,hit,hi,his,he}, 对应的二进制编码为:

c u p
01100011 01110101 01110000

01100011 01110101 01110000
C O P

01100011 01101111 01110000

```

c      o      p      y
01100011 01101111 01110000 01111001

```

h i t

01101000 01101001 01110100
h i

01101000 01101001

```

h      i      s
01101000 01101001 01110011

```

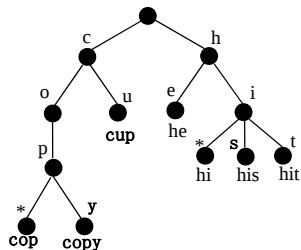
h e

(1) 画出英文字母的字符Trie结构

(2) 画出以其二进制编为分支的PAT Trie结构。

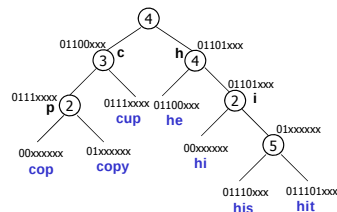
第4题

(1) 参考答案：这道题恰好没有用到压缩



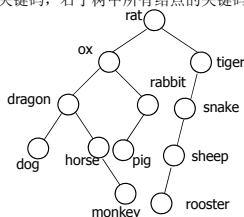
第4题(续)

(2) 参考答案：(PAT Trie)



第5题

1) 二叉搜索树：左子树中所有结点的键码值都小于该结点的键码，右子树中所有结点的键码值都大于该结点的键码。



查找成功的平均搜索长度 = $(1+2*2+3*3+4*4+5*2)/12=10/3$

第5题 (续)

2) AVL树：平衡的二叉搜索树，任何结点的左子树和右子树的高度最多相差1。

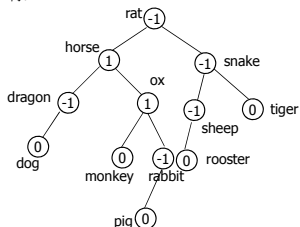
要求：当插入一个结点导致树不平衡时注意找到不平衡节点进行相应的旋转即可；

考点：各种旋转的理解，总共进行了三次旋转，分别在sheep、monkey、dog

(提醒同学们细心，比如有的同学pig分到了rat右子树)

第5题 (续)

2) AVL树：



查找成功的平均搜索长度 = $(1+2*2+3*3+4*4+5)/12=19/6$

第5题 (续)

3) 伸展树：被访问节点经过一系列的双旋转（包括一字形和之字形）直到结点到达根结点或者根结点的子结点。如果需要的话，再进行一次单旋转使结点成为根结点。

4) 半伸展树与伸展树的区别：

a) 一字形情况只进行一次单旋转

b) 已处理过的结点不参与下一轮的伸展，而从其父结点开始展开；

