

# 北京大学信息学院 2005-2006 学年

## 《数据结构与算法》期中答案

——张铭、赵海燕、王腾蛟

### 一、（20 分）填空

1. （3 分）c)
2. （7 分）DBAEHFIGCJ
3. （4 分）33
4. （6 分）共 16 个空格，对了错一个扣 0.5，扣完为止

父结点的下标	-1	0	0	0	0	0	0	6	0	0	0	9	0	0	12	0
结点值	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
结点的下标	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

### 二、（30 分）辨析题

1. （10 分）以下各 5 分

下标	0	1	2	3
P	a	A	b	b
N	0	1	0	0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	a	b	a	b	b	a	a	a	b	b					
A	a	b	b												
			×	i = 3	j=3	n[j-1] = 0	(j=0)								
			a	a	b	b									
			×		I=4	j=1	n[j-1]=0	(j=0)							
				a	a	b	b								
			×		i=4	j=0	i++								
					a	a	b	b							
			×		i=5	j=0	i++								
						a	a	b	b						
								×	i=8	j=2	n[j-1]=1	(j=1)			
							a	a	b	B	√ 匹配				

											成功	
--	--	--	--	--	--	--	--	--	--	--	----	--

2. (10分) 这种说法正确 (1分)。证明如下:

【充分性】6分

(1) 当  $n=3$  时, 满足约束条件 (不存在下标  $i, j, k$ , 满足  $i < j < k$  同时  $P_j < P_k < P_i$ ) 的序列有 5 种: 123、132、213、231、321, 都可以利用栈得到;

(2) 假设命题在元素个数为  $n-1$  时成立, 下面证明命题在元素个数为  $n$  时同样成立:

序列  $P_1, P_2, \dots, P_n$  中, 设  $P_i=n$ 。则  $P_1, \dots, P_{i-1}, P_{i+1}, \dots, P_n$  是由 1, 2,  $\dots, n-1$  组成的排列, 且满足约束条件, 根据假设该序列可以利用栈得到。设  $P_j=n-1$  则:

(a)  $j \leq i-1$  时:  $j \leq i-1$  说明  $n-1$  在弹出  $P_{i-1}$  之前已经压入栈中, 因此当  $P_{i-1}$  从栈中弹出之后, 将  $n$  压入栈中并立即弹出, 之后按照  $P_{i+1}, \dots, P_n$  原先的顺序弹栈, 即可获得  $P_1, P_2, \dots, P_n$ 。

(b)  $j > i+1$  时: 我们首先证明当  $j > i+1$  时,  $j$  一定等于  $i+1$ 。利用反证法: 若  $j > i+1$ , 则在  $n(P_i)$  和  $n-1(P_j)$  之间至少有一个数, 且这个数小于  $n-1$ , 它的下标大于  $i$  小于  $j$ , 这种情况与条件矛盾。因此  $j=i+1$ 。这说明在序列  $P_1, \dots, P_{i-1}, P_{i+1}, \dots, P_n$  的生成过程中, 弹出  $P_{i-1}$  之后, 将会压入  $n-1$  (也就是  $P_{i+1}$ ) 并立即弹出。我们作出修改如下: 在压入  $n-1$  之后再压入  $n$ , 并立即弹出  $n$  和  $n-1$ , 之后按照  $P_{i+2}, \dots, P_n$  的顺序弹栈, 即可获得  $P_1, P_2, \dots, P_n$ 。

因此  $P_1, P_2, \dots, P_n$  可以利用栈得到, 命题在  $k=n$  时同样成立。

【必要性】3分

利用反证法, 如果存在  $P_j < P_k < P_i$ , 则  $P_k$  应比  $P_j$  后压入, 但此时取出的顺序则为  $P_j$  在  $P_k$  之前, 这显然与栈先进先出的原则矛盾, 所以, 同样不存在下标  $i, j, k$ , 满足  $i < j < k$  同时  $P_j < P_k < P_i$ 。  
证毕。

3. (10分)

(1) Huffman 编码 (5分)

A: 0100 B: 0101 C: 110 D: 111 E: 011 F: 10 G: 00

(2) 固定编码外部路径长为 186, Huffman 编码外部路径长为

$$4 * (4+5) + 3 * (6+7+10) + 2 * (18+12) = 165$$

$$165 / 186 * 18600 = 16500 \text{ (位)} \quad // 3 \text{ 分}$$

$$1 - 165/186 = 11.3\% \quad // 2 \text{ 分}$$

### 三、(20分) 算法辨析题

1. (5) 理解树的递归周游

算法 1 错

// 1 分

对有些部分重复访问

// 1 分

把 “`traverse(root->rightchild());`” 改为 `root = root->RightSibling();` //3 分

2. (15 分)

算法错

// 1 分

错误在第 14 行、18 行、20 行（实际的第 19 行）

// 3 分

不要理会学生修改的语法等。

(1) 第 14 行，改为：

// 4 分

```
if (tempparent==NULL) //删除替换结点
    pointer->left=temppointer->leftchild();
else tempparent->right=temppointer->leftchild();
```

或者

```
if (tempparent!=NULL) //删除替换结点
    tempparent->right=temppointer->leftchild();
else pointer->left=temppointer->leftchild();
```

(2) 第 16 行改为：

// 3 分

```
else temppointer = pointer->leftchild();
```

(3) 第 20 行改为：

// 4 分

```
if (parent==NULL)
    root=temppointer;
else if (parent->leftchild() == pointer)
```

## 四、（15 分）算法填空

// ? 1 3 分，各 1.5

```
p=p->RightSibling();
q=q->RightSibling();
```

// ? 2 8 分

```
if (p&&!q || !p&q) return false; //其中一个不空而另一个空, false
if (aQueue.empty()) return true; /* 当队列为空时，说明已经比较完
毕，返回 true。两个队列的大小是相同的，因为进队次数相同，出队次数也相同。
*/
```

或者先断队列空

```
if (aQueue.empty()) return true;
if (p || q) return false; //其中一个不空而另一个空, false
```

// ? 3 4 分

```
p=p->leftMostChild();
q=q->leftMostChild();
```

## 五、（15 分）算法设计

假设根结点处于 0 层，函数 Nodelayer( )返回值为 x 的结点的层次，若无值为 x 的结点，返回-1。

本题没有要求算法代价分析，不必看这个部分。

- (1) 算法思想 2 分；
- (2) 算法框架 12 分；
- (3) 注释 1 分

```
template<class T>
int NodeLayer(BinaryTreeNode<T>*root, <T> x)
{
    int i=1;
    if (!root)
        return -1; // root->element 也可以
    if (root->value() ==x)
        return 0; // return 1; 也算可以
    else
        if ((i= NodeLayer (root->leftchild,x))!= -1)
            return i+1;
        else
            if ((i= NodeLayer (root->rightchild,x))!= -1)
                return i+1;
            else
                return -1;
}
```