

2013 年《数据结构与算法 A》期中考试试题

姓名_____ 学号_____ 任课教师_____ 考场_____

题号	一	二	三	四	总分
得分					

注意事项:

1. 全部题目都在空白答题纸上解答。
2. 本试卷对算法设计都有质量要求, 请尽量按照试题中的要求来写算法。否则将酌情扣分。
3. 请申明所写算法的基本思想, 并在算法段加以恰当的注释。

一、 填空(28 分)

1. (2 分) 下面函数的时间复杂度是_____。

```
int foo(int n) {  
    int s = i = 0;  
    while (s < n) { i++; s += i; }  
    return s;  
}
```
2. (2 分) 二叉树的叶结点在前序、中序、后序的遍历序列中的相对次序:_____。
A. 都不相同;
B. 完全相同;
C. 前序和中序相同, 而与后序不同;
D. 中序和后序相同, 而与前序不同。
3. (2 分) 对于非空满 K 叉树, 其分支结点的数目为 n, 则其叶结点的数目为_____。
4. (2 分) 在_____中, 即使丢失了头结点, 只要指出表中任何一个结点的指针, 也可以访问到该结点的前驱结点。
A. 线性单链表 B. 双向链表 C. 线性链表 D. 循环链表
5. (2 分) 若森林 F 对应的二叉树 B 中有 m 个结点, B 的根结点 r 的右子树具有 n 个结点, 那么森林 F 中第 1 棵树的结点数为_____。
A. m-n; B. m-n-1; C. n+1; D. 无法确定。
6. (3 分) 给定数据序列为 {a, b, c, d, e, f, g} 的输入流、以及一个空队列, 每步可进行以下两种操作: (1) 取出输入流的下一个数据、入队列, 或 (2) 一个数据出队列、输出。当输入流和队列均为空时, 输出序列不可能是以下哪些序列: _____。
A. {d, e, c, f, b, a} B. {f, e, g, d, a, c, b}

C. {e, f, d, g, b, c, a} D. {c, d, b, e, f, a, g}

7. (2分) 对一棵完全 k 叉树, 按照广度优先周游顺序给结点从左到右依次连续编号, 第一个结点编号为 0, 则编号 m ($m \neq 0$) 的结点的父结点编号是_____。

8. (4分) 若一棵二叉树中结点值在 1 到 1000 之间, 现在要在其中查找值为 363 的结点。下面序列中_____不是查找过的序列?

- A. 2, 252, 401, 398, 330, 344, 397, 363;
- B. 924, 220, 911, 244, 898, 258, 362, 363;
- C. 925, 202, 911, 240, 912, 245, 363;
- D. 2, 399, 387, 219, 266, 382, 381, 278, 363。

9. (6分) 使用重量权衡合并规则与路径压缩, 对下列从 0 到 15 之间的数的等价对进行归并。在初始情况下, 集合中的每个元素分别在独立的等价类中。当两棵树规模同样大时, 使结点数值较大的根结点作为值较小的根结点的子结点。

(0,2) (1,2) (3,4) (3,1) (3,5) (9,11) (12,14) (3,9) (4,14) (6,7) (8,10) (8,7) (7,0) (10,15) (10,13)

请填写下面表格的空白部分树的父指针表示法的数组表示。也就是所有等价对都被处理后, 所得父结点的下标值 (没有父结点则填 “-1”)。

父结点下标																
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

10. (3分) 使用栈计算逆波兰表达式 (操作数均为一位数) $1\ 2 + 4 * 5 + 3 -$, 当处理到数字 5 时, 栈的内容 (以栈底到栈顶从左往右的顺序书写) 为_____。

二、 辨析与简答(共 3 题, 共 27 分)

1. (8分) 序列 23, 17, 14, 6, 13, 10, 1, 5, 8, 12 是否为一个最大值堆? 若是, 请说明理由, 否则请严格按照筛选法建堆的过程将其调整成为最大值堆, 并画出调整建堆的逐步过程。

2. (9分) 已知某电文中共出现了 10 种不同的字母 (A, B, C, D, E, F, G, H, I, J), 每种字母出现的相对频率分别为 6、8、5、3、7、22、10、9、1、40, 现在对这段电文用三进制进行非定长前缀编码 (码字由 0、1、2 组成), 问电文编码总长度至少有多少位? 请画出相应编码方案的图示。

3. (10分) 以下是计算模式 P 的 next 向量的算法

(1) 请在空缺处填写相应的语句, 使得算法完整;

4. 请填补完整; 并用此方法计算模式 $P = \text{“abcdaabcb”}$ 的 next 向量。

```
int *findNext(string P) {
```

```

int i = 0;
k = -1;
int m = P.length();
int* next = new int[m];
next[0] = -1;
while (i < m) {
    while (k >= 0 && P[i] != P[k])
        k = next[k];
    i++;
    k++;
    if (i == m) break;
    // 待填入空缺代码
}
return next;
}

```

(2) 采用上述算法，计算模式P= “abcdaabcb” 的next向量。

三、 算法填空(每空 2 分, 共 20 分)

1. 完成非递归的二叉树搜索算法，给定一个二叉树（不是BST）和一个值K，如果K出现在二叉树中则返回true，否则返回false。

```

template<class T>
struct BinaryTreeNode {
    T value();
    BinaryTreeNode<T>* leftchild();
    BinaryTreeNode<T>* rightchild();
};

template<class T>
bool search( BinaryTreeNode<T>* root , T k ) {
    using std::stack;
    stack< BinaryTreeNode<T>*> aStack;
    BinaryTreeNode<T>* pointer = root ;
    while ( _____ // 填空1 ) {
        if ( pointer ) {
            _____ // 填空2
            pointer = pointer->leftchild() ;
        } else {

```

```

        _____ // 填空3
        if ( _____ // 填空4 )
            return true;
        pointer = pointer->rightchild();
        _____ // 填空5
    }
}
return false;
}

```

2. 下面的算法将一个用带度数的后根次序法表示的森林转换为左子结点/右兄弟结点法表示。请利用题目给出的树结点ADT和栈ADT，填充算法的空格，使其成为完整的算法。空格中可能需要填写0到多条语句（或表达式）。

```

template<class value_type>
class stack {
public:
    bool empty();                // 判断栈空
    int size();                  // 返回栈大小
    value_type &top();            // 读栈顶
    void push(const value_type& X); // 入栈
    void pop();                  // 出栈
};

struct Node<T> {
    T info    // 结点的数据信息
    int degree //结点的度数信息
};

template<class T>                // 树结点类
class TreeNode {
public:
    bool isLeaf();                // 判断当前结点是否为叶结点
    T Value();                    // 返回结点的值
    TreeNode<T> *LeftMostChild(); // 返回第一个左孩子
    TreeNode<T> *RightSibling();  // 返回右兄弟
    void setValue(const T&);      // 设置当前结点的值
    void setChild(TreeNode<T> *pointer); // 设置左孩子
    void setSibling(TreeNode<T> *pointer); // 设置右兄弟
};

TreeNode<T> *Convert(Node* nodes, int size) {
    TreeNode<T> *cur, *temp1, *temp2;
    stack<TreeNode<T>*> Cstack;

```

```

for ( int i = 0; i < size; i++) {
    cur = new TreeNode<T>(nodes[i].info);
    if ( nodes[i].degree == 0 )
        _____ // 填空 1
    else {
        assert( nodes[i].degree <= Cstack.size() );
        temp2 = NULL;
        for (int j = 0; _____ // 填空 2; j++) {
            temp1 = Cstack.top();
            Cstack.pop();
            _____ // 填空 3
            temp2 = temp1;
        }
        _____ // 填空 4
        Cstack.push(cur);
    }
}
cur = temp2 = NULL;
while( !Cstack.empty() ) {
    cur = Cstack.top();
    Cstack.pop();
    _____ // 填空 5
    temp2 = cur;
}
return cur;
}

```

四、 算法设计与实现 (25 分)

注意：(1) 对算法设计有质量要求，请尽量要求写出高效算法（做算法分析，否则将酌情扣分）；
 (2) 请申明所写算法的基本思想，并在算法段（C++伪代码）加以恰当的注释。

- （12分）设计算法来判断一个给定的二叉树是否为完全二叉树，并分析算法的时间复杂度。
- （13分）请利用两个栈S1和S2来模拟一个队列。已知栈的三个运算定义如下： PUSH(ST,x): 元素x入ST栈； POP(ST,x): ST栈顶元素出栈，赋给变量x； EMPTY(ST): 判ST栈是否为空。

(1) 利用栈的运算来实现该队列的三个运算

enqueue: 插入一个元素入队列；

dequeue: 删除一个元素出队列；

queue_empty: 判队列为空。

(2) 给定kn个元素，按顺序进行n次enqueue、n次dequeue操作，重复k次，分别求出栈S1和S2的三种运算的操作次数。