

装订线内

不要答题

北京大学信息科学技术学院考试试卷

考试科目：数据结构与算法 A 姓名：_____ 学号：_____

考试时间：2016 年 11 月 9 日 任课教师：_____

题号	一	二	三	四	五	总分
分数						
阅卷人						

北京大学考场纪律

- 1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。
 - 2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机等）不得带入座位，已经带入考场的必须放在监考人员指定的位置，并关闭手机等一切电子设备。
 - 3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束监考人员宣布收卷时，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。
 - 4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号，不准携带与考试内容相关的材料参加考试，不准抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有严重违纪或作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学习纪律管理规定》及其他相关规定严肃处理。
 - 5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。
- 学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

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。

以下为试题和答题纸，共 5 页，请把答案写在答题纸。

得分

一、 选择(每题 2 分, 共 20 分)

1. 下面函数的时间复杂度是_____。

```
int work(int n, int m) {
    int sum = 0;
    for (int i = 1; i <= m; i *= 2)
        for (int j = 1; j <= n; j *= 2) {
            sum += i * j;
        }
    return sum;
}
```

2. 下面关于线性表的叙述中, 正确的是哪些?

- A. 采用顺序存储的线性表, 必须占用一片连续的存储单元。
- B. 采用顺序存储的线性表, 便于进行插入和删除操作。
- C. 采用链接存储的线性表, 不必占用一片连续的存储单元。
- D. 采用链接存储的线性表, 便于插入和删除操作。
- E. 采用链接存储的线性表, 插入第 i 个元素的时间与 i 的数值成正比。
- F. 采用链接存储的线性表, 在已知的结点 p 后插入一个新节点的时间复杂度与结点 p 的位置有关。

3. 若元素 a 、 b 、 c 、 d 依次进栈, 则可能的出栈序列有_____种; 若有元素 a 、 b 、 c 、 d 、 e 依次进栈, 则可能的出栈序列有_____种。

4. 一个字符串中_____称为该串的子串。abcaab 的不相等真子串个数为_____。

5. 一个具有 n 个结点的二叉树的叶结点的个数最多为_____。

6. 一个有 5 层结点的完全 3 叉树。按前序遍历周游给结点从 1 开始编号, 则第 100 号结点的父结点的编码为_____。(注释: 独根树为 1 层的)

7. 将序列 {12, 70, 33, 65, 24, 56, 48, 92, 86, 33} 按建堆方法调整为小值堆, 调整后的序列为_____。

8. 假设用于通信的电文仅由 8 个字符 { a, b, c, d, e, f, g, h } 组成, 字符在电文中出现的频率分别为 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10, 对这些字符进行哈夫曼编码的外部路径长度为_____。

9. 将森林 F 转换为对应的二叉树 T , F 中的叶结点的个数为_____。

- A. T 中叶结点的个数
- B. T 中度为 1 的结点个数
- C. T 中左子指针为空的结点个数
- D. T 中右子指针为空的结点个数

10. 一棵树 T 有 20 个度为 4 的结点, 10 个度为 3 的结点, 1 个度为 2 的结点, 10 个度为 1 的结点, 则树 T 的结点个数为_____。

得分

二、 辨析与简答(共 32 分)

1. (6分)请谈谈循环队列和链表的优缺点。当遇到如下情况时,应当使用哪一种数据结构?

情况:你要维护食堂鸡腿饭的排队队伍,队伍有两种操作:1.排在队首的人出来打鸡腿饭后离开;2.一个新来的人排在队尾。你知道食堂不太大,所以队伍的最大长度你可以估计出来。

2. (4分)如何找出具有相同的中序序列和后序序列的所有二叉树?
3. (4分)从空二叉树开始,将关键码{18,73,10,5,68,99,27,41,51,32,25}严格按照BST(二叉搜索树)的插入算法逐个插入BST,画出这样构造出的BST树,并写出对该BST按照前序遍历得到的序列。

4. (6分)以下是计算模式P的next向量的算法:

```
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;
        if (P[i] == P[k])
            next[i] = next[k];
        else
            next[i] = k;
    }
    return next;
}
```

采用上述算法,计算模式P="abcdaabcbab"的next向量。

i	0	1	2	3	4	5	6	7	8	9
P[i]	a	b	c	d	a	a	b	c	a	b
next[i]										

5. (6分)数组{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,并画出逐步调整建堆的过程。
6. (6分)对下列15个等价对进行合并,给出所得等价类树的图示。在初始情况下,集合中的每个元素分别在独立的等价类中。使用重量权衡合并规则,合并时子树结点少的并入结点多的那棵(多的那个作为新树根,少的那个根作为新根的直接子结点);若两棵树规模同样大,则把根值较大的并入根值较小(新树根取值小的)。同时,采用路径压缩优化。

(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)

得分

三、 算法填空(每空 2 分, 共 16 分)

1. 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>
class TreeNode {                // 树结点类
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(); // 返回第一个子结点
    TreeNode<T> *RightSibling(); // 返回右兄
    void setValue(const T&);      // 设置当前结点的值
    void setChild(TreeNode<T> *pointer); // 设置左子
    void setSibling(TreeNode<T> *pointer); // 设置右兄
};

template<class T>
TreeNode<T> *Convert(Node<T>* nodes, int size) {
    TreeNode<T> *cur, *templ, *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;
}

```

2. 给定一个二叉树的根结点和树中任意两个结点，补全下面的程序段以求这两个结点的最近公共祖先。

```

TreeNode* lowestCommonAncestor(
    TreeNode* root, TreeNode* p, TreeNode* q) {
    if ( // 填空 6 ) return root;
    TreeNode* left = lowestCommonAncestor(root->left, p, q);
    TreeNode* right = lowestCommonAncestor(root->right, p, q);
    if ( // 填空 7 ) return root;
    return // 填空 8 ? left:right;
}

```

得分

四、 算法设计与实现（16 分）

1. （8 分） 现有两个满足先进先出原则的队列 A 和 B，在队列上可进行以下 4 个操作：

front()：获得队首元素
 push_back(T x)：把元素 x 插到队尾。
 pop_front()：删除队首元素。
 empty()：判断是否为空。

请用 A 和 B 模拟一个栈的 4 个操作：top()，pop()，push()，empty()。只要给出实现栈操作的基本思想，对时间效率无具体要求。

2. （8 分） 对于两个串 A 和 B，给出一种算法使得能够保证正确地求出最大的 L 值，使得 A 串中长为 L 的前缀与 B 串中长为 L 的后缀相等。

得分

五、 分析证明题（共 16 分）

1. （6 分）试证明在一棵树中， u 是 v 的祖先，当且仅当在先序遍历中 u 在 v 之前且在后序遍历中 u 在 v 之后。
2. （10 分）对于满二叉树：
 - （1） 试问具有 n 个叶结点的树中共有多少个结点？
 - （2） 试证明： $\sum_{i=1}^n 2^{-(l_i-1)} = 1$ ，其中 n 为叶结点的个数， l_i 表示第 i 个叶结点所在的层次（设根结点所在的层次为 0）。