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：2017 年 11 月 15 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. 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、 选择与填空（每空 2 分，共 24 分）

1. 下面函数的时间复杂度是（ ）

```
void recursive(int n, int m, int k){
    if (n <= 0)
        printf("%d, %d\n", m, k);
    else {
        recursive(n-1, m+1, k);
        recursive(n-1, m, k+1);
    }
}
```

A. O(n*m*k) B. O(n^2*m^2)

B. O(2^n) D. O(n!)

2. 完成在双循环链表结点 p 之后插入 s 的操作为( ):

A. p->next->prev=s; s->prev=p; s->next=p->next; p->next=s;

B. p->next->prev=s; p->next=s; s->prev=p; s->next=p->next;

C. s->next=p->next; s->prev=p; p->next->prev=s; p->next=s;

D. s->prev=p; s->next=p->next; s->prev->next=s; s->next->prev=s;

3. 设栈 S 和队列 Q 初始状态为空，元素 e1，e2，e3，e4，e5，e6 依次通过栈 S，一个元素出栈后即进队列 Q，若 6 个元素出队序列是 e2，e4，e3，e6，e5，e1，则栈 S 的容量至少是( )

A.2; B.3; C.4; D.6

4. 设循环队列的容量为 40 （序号从 0 到 39），现经过一系列的入队和出队运算后： 1）front=12，rear=19；2）front=19，rear=12；在这两种情况下，循环队列中的元素个数分别为_____和_____。

5. 一个 n 位的字符串共有_____个子串。

6. 已知二叉树有 n 个结点 (n > 0)，则度为 2 的结点最多有_____个。

7. 用数组存储一个有 50 个元素的最小值堆。已知堆中没有重复元素。给出最大元素的下标可能的取值范围为 _____ 到 _____。

8. 假设用于通信的电文由 5 个字符组成。已知其中一个字符的 Huffman 编码为 1101，则该 Huffman 编码树的平均编码长度为_____。

9. 假设先根次序遍历某棵树的结点次序为 GABCDIJEFH，后根次序遍历该树的结点次序为 BADIJCFEHG，那么 I 结点的父结点是_____。

10. 一个拥有 144 个结点的完全三叉树，现在从倒数第二层上任取一个结点，该结点为叶结点的概率是_____。（请用化简后的分数表示）。

得分

二、 辨析与简答(共 24 分)

1. （6 分）现有一个由单链表和循环单链表构成的特殊链表，单链表的头元素是 head，末尾元素为 tail，head->…->tail 即是一条普通的链表，同时 tail 元素还是另一条循环单链表的头元素。如 A->B->C->D->E->F->G->H->E 就是一个特殊链表，其中 head 为 A，tail 为 E。

   现在你只知道 head，但是你想知道这个循环链表中有多少元素，但同时你的剩余空间十分的小，所以你想用尽可能小的空间来解决问题，请问你会怎么做？（依据占用空间给分）

2. （6 分）假设以 I 和 O 分别表示入栈和出栈操作，栈的初始和终态均为空，入栈和出栈的操作序列可表示为仅由 I 和 O 组成的序列，称可以操作的序列为合法序列，否则为非法序列。如 IOIIOIOO 合法，IOOIOIIO 和 IIIOIOIO 为非法。请给出一个算法，判断所给操作序列是否合法。

3. （6 分）给定一棵用数组存储的完全二叉树{10, 5, 12, 3, 2, 1, 8}

   （1）用筛选建堆法将该完全二叉树调整为最大值堆。画出调整后得到的最大值堆，并说明在调整过程中都依次交换了哪些元素。（注：画堆只需要写出层次遍历序列结果即可）

   （2）将 6、7、14 按顺序插入（1）得到的最大值堆，堆的空间足够大，画出插入 14 后得到的最大值堆，并说明在插入 14 的过程中都依次交换了哪些元素。

   （3）对（2）得到的堆进行 3 次 deleteMax，画出第 3 次 deleteMax 后得到的堆，并说明在第 3 次 deleteMax 的过程中都依次交换了哪些元素。

4. （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数多的那棵（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) (12,9) (4,14) (6,7) (8,10) (8,7) (7,11) (10,15) (10,13)

得分

三、 算法填空(每空 3 分，共 24 分)

1. 下面的算法利用一个栈将一给定的序列从小到大排序。长度为n的序列由1, 2, …, n这n个连续的正整数组成。请补全下面的代码段，使其可以判断给定的序列是否可以利用一个栈进行排序，如果可以，输出相应的操作。

   例如：序列 4 3 1 2，此序列可以排序，输出：push push push pop push pop pop pop。

```
template <class T>                          // 栈的元素类型为 T
class Stack {
    public:
        void clear();                       // 变为空栈
        void push(const T item);            //  item 入栈
        T pop();                            // 返回栈顶内容并弹出
        T top();                            // 返回栈顶内容但不弹出
        bool isEmpty();                     // 若栈已空返回真
        bool isFull();                      // 若栈已满返回真
};
```

```
Stack<int> s;                                   //   s 为栈

int sequence[maxLen],   len;                    // sequence 为待排序序列，  len 为序列长度
bool islegal()   {                              // 判断序列是否可以排序
    int i, j, k;
    for (i = 0; i < len; i++) {
        for (j = i+1; j < len; j++) {
            for (k = j+1; k < len; k++) {
                if (   //填空 1    )
                    return false;
            }
        }
    }
    return true;
}
void transform()   {                            // 输出转换操作
    int cur = 1,   i = 0;
    while   (   //填空 2   ) {
        while (s.isEmpty()   ||   //填空 3  ) {
            //填空 4
            cout << "push ";
            if (cur == s.top()) break;
        }
        s.pop();
        cout << "pop ";
        ++cur;
    }
}
```

2. 下面的算法将一个用带右链的先根次序法表示的森林转换为用带度数的后根次序法表示。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。

```
template<T>          // 数据结构
class RlinkTreeNode {
    int ltag;                       // 左标记
    T info;
    RlinkTreeNode<T> *rlink; // 右链
}
template<T>
class PostTreeNode {
    T info;
    int degree;                     // 度数
}
// 算法描述：递归遍历用带右链的先根次序法表示的森林，同时计算出度数并转成后根次
        序森林；设带右链的先根次序法表示的森林为数组
```

```
RlinkTreeNode RlinkTree[n];
// 带度数的后根次序法表示的森林为数组
PostTreeNode PostTree[n];
int count = 0;                          // 计数器
int convert(PostTreeNode * node) { // 返回值表示所有右兄弟的数量（包括自己）
    int tmp;
    if (1 == node->ltag) {              // 若没有左子结点，将该结点输出到 PostTree
        ____// 填空 5____
        PostTree[count].degree = 0; // 度数必然是 0
        count++;
    }
    else {                              // 有左子结点则压栈
        tmp = convert(node+1);          // 访问第一个子结点，并返回度数
        PostTree[count].info = node->info;
        ____// 填空 6____
        count++;
    }
    if (____// 填空 7____) {   // 有右兄弟(，访问并返回右边兄弟的数量+1)
        return convert(node->rlink) + 1;
    }
    else {                              // 没有右兄弟
        ____// 填空 8____
    }
}
```

得分

四、 算法设计与实现 (14 分)

1. （8 分） 编写算法伪码，对给定的字符串 str，返回其最长重复子串及其下标位置。如 str=”abcdaccdac”，则子串”cdac”是 str 的最长重复子串，下标为 2。（重复子串允许重叠）

2. （6 分）给出一个算法，求一棵树的树高。可以写出伪代码，需要相应的注释，或者写出详细算法思路。对时间复杂度无具体要求。

得分

五、 分析证明题 (共 14 分)

1. （8 分）二叉树的内部路径长度是指所有节点的深度的和。假设将 N 个互不相同的随机元素插入一棵空的二叉搜索树。请证明得到的二叉搜索树的内部路径长度期望为 O(NlogN)。

2. （6 分）证明按先根次序周游树获得的序列与其对应二叉树的前序周游获得的序列相同。