

# 上海交通大学试卷(A卷)

(2017至2018学年第\_一学期)

班级号 \_\_\_\_\_ 学号 \_\_\_\_\_ 姓名 \_\_\_\_\_

课程名称 \_\_\_\_\_ 数据结构B类 成绩 \_\_\_\_\_

## 一、选择题(每题1.5分,共15分)

1. 在一个单链表中,每个结点有一个指向后继结点的指针域next。若删除p指向的后继结点,则应该执行\_\_\_\_\_。  
A.  $p->next = p->next->next$       B.  $p = p->next; p->next = p->next->next$   
C.  $p = p->next$       D.  $p = p->next->next$
2. 设有5000个待排序的数字,如果需要用最快的方法选出其中最小的10个,则用下列哪个方法可以达到此目的\_\_\_\_\_。  
A. 快速排序      B. 堆排序  
C. 归并排序      D. 插入排序
3. 设n是描述问题规模的正整数,下列程序片段的时间复杂度是\_\_\_\_\_。  

```
y = 0;  
while (n >= (y+1) * (y+1)) y++;
```

  
A.  $O(\log n)$       B.  $O(n)$       C.  $O(n \log n)$       D.  $O(\sqrt{n})$
4. 高度(根的高度为1)为50的二叉树上只有度为0和度为2的结点,则此类二叉树中所包含的结点数最少为\_\_\_\_\_。  
A. 50      B. 59      C. 99      D. 100
5. 对一组数据(84, 47, 25, 15, 21)排序,数据的排列次序在排序的过程中的变化为:  
(1) 84, 47, 25, 15, 21    (2) 15, 47, 25, 84, 21    (3) 15, 21, 25, 84, 47    (4) 15, 21, 25, 47, 84  
则采用的排序算法是\_\_\_\_\_。  
A. 选择      B. 冒泡      C. 快速      D. 插入
6. 设森林F对应的二叉树为B,它有m个结点,B的根为P,P的右子树结点个数为n,森林F中的第一棵树的结点个数是\_\_\_\_\_。  
A.  $m-n$       B.  $m-n-1$       C.  $n+1$       D. 无法确定
7. 希尔排序的增量序列必须是\_\_\_\_\_。  
A. 递增的      B. 递减的      C. 随机的      D. 非递减的
8. 为解决计算机与打印机之间速度不匹配的问题,通常设置一个打印数据缓冲区,主机将要输出的数据依次写入该缓冲区,而打印机则依次从该缓冲区中取出数据。该缓冲区的逻辑结构应该是\_\_\_\_\_。  
A. 栈      B. 队列      C. 树      D. 图

我承诺，我将严格遵守考试纪律。

承诺人：\_\_\_\_\_

题号	一	二	三	四	五	六	总分
得分							
批阅人(流水阅卷教师签名处)							

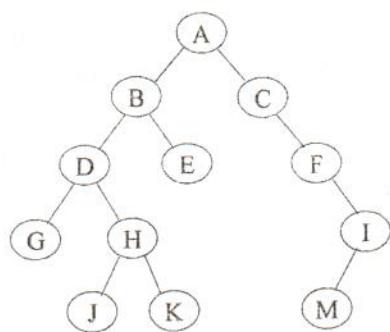
9. 设栈 S 和队列 Q 的初始状态均为空, 元素 abcdefg 依次进入栈 S。若每个元素出栈后立即进入队列 Q, 且 7 个元素出队的顺序是 bdcfeag, 则栈 S 的大小至少是\_\_\_\_\_。  
A. 1      B. 2      C. 3      D. 4
10. 已知关键序列 5, 8, 12, 19, 28, 20, 15, 22 是最小化堆, 插入关键字 3, 调整后得到的最小化堆是\_\_\_\_\_。  
A. 3, 5, 12, 8, 28, 20, 15, 22, 19      B. 3, 5, 12, 19, 20, 15, 22, 8, 28  
C. 3, 8, 12, 5, 20, 15, 22, 28, 19      D. 3, 12, 5, 8, 28, 20, 15, 22, 19

二、填充题（每空 1.5 分，共 15 分）

1. 对于顺序存储结构的栈, 假如有足够的空间, 其进栈和出栈的时间复杂度均为\_\_\_\_\_。
2. 设用于通信的电文仅由 8 个字母组成, 字母在电文中出现的频率分别为 7, 19, 2, 6, 32, 3, 21, 10, 根据这些频率作为权值构造哈夫曼树, 则这棵哈夫曼树的高度为\_\_\_\_\_。
3. 已知一个后缀表达式为 5, 1, 2, +, 4, \*, +, 3, -, 请写出其中缀表达式\_\_\_\_\_。
4. 散列表的地址范围为 0~32, 散列函数  $H(k) = k \% 31$ 。采用线性探测法处理冲突, 将关键字序列 26, 25, 75, 38, 18, 58, 76 依次存储到散列表中。元素 76 存放在散列表中的地址是\_\_\_\_\_。
5. 高度 (根的高度为 1) 为 k 的完全二叉树至少有\_\_\_\_\_个结点, 至多有\_\_\_\_\_个结点。
6. 用 P 表示进栈操作, D 表示出栈操作, 若元素进栈的顺序为 abcde, 为了得到 bcade 出栈顺序, 相应的 P 和 D 的操作串为\_\_\_\_\_。
7. 给出一组关键字  $T = (20, 4, 34, 5, 16, 33, 18, 29, 2, 40, 7)$ , 要求从小到大进行排序, 试给出快速排序 (选第一个记录为标准元素) 第一趟划分结果为\_\_\_\_\_。
8. 若待排序列已基本有序, 要使它完全有序, 则从关键码比较和移动次数考虑, 应当使用的排序方法是\_\_\_\_\_。
9. 若查找每个记录的概率均等, 则在具有 n 个记录的连续顺序文件中采用顺序查找法查找一个记录, 其平均查找长度 ASL 为\_\_\_\_\_。

三、简答题（每题 5 分，共 10 分）

1. 画出下列按左孩子右兄弟方式转化的二叉树所对应的森林。



2. 写出下面算法的功能并分析复杂度。数组 visit 是一个 bool 类型的全局变量，且初值为 false。

```
struct graph {
    int vexnum, arcnum;
    char vexs[N];
    int arcs[N][N];
};

void xx (int i, graph *g) {
    int j;
    cout << "node:" << g->vexs[i] << endl;
    visited[i] = true;
    for (j = 0; j < g->vexnum; j++)
        if ((g->arcs[i][j] == 1) && (!visited[j])) xx(j, g);
}
```

#### 四、分析题（15分）

1. 已知用线性有序链表存储整数集合的元素，阅读下面算法，并回答下列问题：

```
int ABC (LinkList ha, LinkList hb) {
    // LinkList 是带有头结点的单链表
    // ha、hb 分别指向存储两个有序整数集合的链表的头指针
    LinkList pa,pb;
    pa = ha->next;
    pb = hb->next;
    while (pa && pb && pa->data == pb->data) {
        pa = pa->next;
        pb = pb->next;
    }
    if (pb != NULL) return 0;
    else return 1;
}
```

- 1) 写出执行程序 ABC (a, b) 的返回值，其中 a 和 b 分别指向存储集合 {2, 4, 5, 7, 9, 12} 和 {2, 4, 5, 7, 9} 的链表的头指针。（3分）
- 2) 简述程序 ABC 的功能。（2分）
- 3) 写出程序 ABC 的时间复杂度。（4分）

2. 已知二叉查找树类 BST 和结点类 BstNode 定义如下：

```
template <class Type> class BST;
template <class Type> class BstNode {                                // 搜索树的结点类
    friend class BST <Type>;
public:
    BstNode () : lchild (NULL), rchild (NULL) {}                  // 构造函数
    BstNode (const Type& d, BstNode <Type>* l = NULL, BstNode <Type>* r = NULL):
        data (d), lchild (l), rchild (r) {};
    ~BstNode () {}                                              // 析构函数
private:
    Type data;                                                 // 数据域
    BstNode <Type> *lchild, *rchild;                            // 指向子女的指针
};

template <class Type> class BST {                                     // 搜索树的类定义
public:
    BST () : root (NULL) {}                                      // 构造函数
    BST (const Type& value);
    ~BST () {}                                              // 析构函数
    void MakeEmpty () {MakeEmpty (root); root = NULL;}          // 清空
    .....
    void DeleteBst (const Type &x);
    .....
private:
    BSTNode <Type>* root;                                       // 指向根结点的指针
}
```

请分别写出下列 BST 类的成员函数 A 和 B 的功能。(6 分)

```
template <class Type>
void BST <Type>::A (const Type &x) {
    BSTNode <Type> *p;

    while (root != NULL && root->data != x) {
        p = root;
        if (root->data < x) {
            root = root->rchild;
            B (p->lchild);
            delete p;
        }
        else {
            root = root->lchild;
            B (p->rchild);
            delete p;
        }
    }
}
```

```
template <class Type>
void BST <Type>::B (BSTNode <Type> * t) {
    if (t) {
        B (t->lchild);
        B (t->rchild);
        delete t;
    }
}
```

五、程序填充题（每空格 2 分，共 20 分）

1. 下列程序是二叉查找树的插入函数的实现，请在空白处填上适合的代码。已知二叉查找树结点的数据成员有：data、left、right，指向二叉查找树根结点的指针是 root，请在空白处填上适合的代码。

```
template <class T>
void BinarySearchTree <T>::insert (const T &x) {
    _____;
}

template <class T>
void BinarySearchTree <T>::insert (const T &x, BinaryNode * &t) {
    if (t == NULL) t = new BinaryNode (x, NULL, NULL);
    else if (x < t->data) _____;
    else if (x > t->data) _____;
}
```

2. 已知下列二叉查找树的非递归查找程序，请在空白处填上适合的代码。二叉查找树结点的数据成员有：data、left、right，指向二叉查找树根结点的指针是 root

```
template <class ElemtType>
bool BinarySearchTree < ElemtType >::Find (const ElemtType & item) {
    BinaryNode *p = root;
    while (p && p->data != item)
        if (item < p->data) _____
        else _____;
    if (_____ ) return true; else return false;
}
```

3. 已知二叉树定义如下：

```
class BinaryTree;
class TreeNode {
    friend class Tree;
    public:
        .....
    private:
        TreeNode * LeftChild, * RightChild;
        int Data;
};
class Tree {
    public:
        Tree ();
        ~Tree ();
        .....
        int childnumber (TreeNode * current);
        .....
};
```

下列函数 int BinaryTree::childnumber (TreeNode \* current), 确定二叉树中的某个结点 current 的子孙个数, 请在空白处填上适合的代码。

```
int BinaryTree::childnumber (TreeNode * current) {
    if (current == NULL) return 0;
    else {
        int number = 0;
        if(_____) _____;
        if(_____) _____;
        return number;
    }
}
```

#### 六、编程题 (25 分)

1. 已知一个带有表头结点的单链表, 结点结构为:

data	next
------	------

假设该链表只给出了头指针 list。在不改变链表的前提下, 请设计一个尽可能高效的算法, 查找链表中倒数第 k 个位置上的结点 (k 为正整数)。若查找成功, 算法输出该结点的 data 值, 并返回 1; 否则, 只返回 0。要求:

- 1) 描述算法的基本设计思想。(4 分)
  - 2) 根据设计思想, 采用程序设计语言描述算法 (使用 C++ 语言实现)。(8 分)
  - 3) 分析算法的时间复杂度。(3 分)
2. 冒泡排序算法是把大的元素向上移 (气泡的上浮), 也可以把小的元素向下移 (气泡的下沉) 请写出上浮和下沉过程交替进行的冒泡排序 (即双向冒泡排序法) 的函数 void Bi-BubbleSort (int a[], int size)。(10 分)