

上海交通大学试卷(A卷)

(2014至2015学年第一学期)

班级号 _____ 学号 _____ 姓名 _____
课程名称 _____ 《数据结构(A类)》 成绩 _____

一、单项选择题(每格1.5分,共24分)

1. 在双向循环链表中,在p指针所指向的结点前插入一个指针q所指向的新结点,其修改指针的操作是 _____. 备注: 双向链表的结点结构为(prev, data, next).
A. $p \rightarrow prev = q; q \rightarrow next = p; p \rightarrow prev \rightarrow next = q; q \rightarrow prev = q;$
B. $p \rightarrow prev = q; p \rightarrow prev \rightarrow next = q; q \rightarrow next = p; q \rightarrow prev = p \rightarrow prev;$
C. $q \rightarrow prev = p \rightarrow prev; q \rightarrow next = p; p \rightarrow prev = q; p \rightarrow prev = q;$
D. $q \rightarrow next = p; q \rightarrow prev = p \rightarrow prev; p \rightarrow prev \rightarrow next = q; p \rightarrow prev = q;$
2. 当图中各条边的权值 _____. 时,广度优先搜索算法可用来解决单源最短路径问题。
A. 不一定相等 B. 均互不相等 C. 均相等 D. 对权值无特别要求
3. 若以一个大小为6的数组来实现循环队列,当前的rear和front的位置分别为0和3,当依次执行:
两次出队、两次入队、一次出队后,当前rear和front位置分别是 _____.
A. 5, 1 B. 2, 5 C. 2, 0 D. 4, 0
4. 如果某函数 $f(N)$ 是 $\Theta(N \log_2 N)$ 的,且函数 $g(N) = f(N^2)$,则 $g(N)$ 是 _____. 的。

A. $\Omega(N^{2.618})$ B. $\Theta(N^2(\log_2 N)^2)$
C. $\Theta(N(\log_2 N)^2)$ D. $O(N^2 \log_2 N)$
5. 以下时间复杂度, _____. 不是 $O(N \log_2 N)$.
A. 快速排序的最坏情况 B. 堆排序的平均情况
C. 归并排序的最坏情况 D. 归并排序的最好情况
6. Dijkstra 算法不适用于寻找以下 _____. 种图的最短路径。
A. 所有边权重均为1的有向图 B. 边的权重均为正数的有向图
C. 边的权重均为正数的无向图 D. 边的权重可能出现负数的有向图
7. 在最大化堆中,不正确的是 _____.
A. 任何节点的数值不超过其父亲 B. 兄弟节点之间具有确定的大小关系
C. 节点的数值不超过其任何一个祖先 D. 整个堆中的最大元素在根节点
8. 某二叉树有N个节点,高度为h,在其中插入一个新的节点,高度发生改变的节点个数最多为 _____.
A. $O(1)$ B. $O(N)$ C. $O(h)$ D. $O(h \log_2 N)$

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. 由两个栈共享一个向量空间的好处是 _____。
A. 减少存取时间，降低下溢发生的机率 B. 节省存储空间，降低上溢发生的机率
C. 减少存取时间，降低上溢发生的机率 D. 节省存储空间，降低下溢发生的机率
10. 对于线性表(7, 34, 55, 25, 64, 46, 20, 10)进行散列存储时，若选用 $H(K) = K \% 9$ 作为散列函数，则散列地址为 1 的元素有 _____ 个。
A. 1 B. 2 C. 3 D. 4
11. 设一组初始记录关键字序列为(50, 40, 95, 20, 15, 70, 60, 45)，则以增量 $d = 4$ 的一趟希尔排序结束后前 4 条记录关键字为 _____。
A. 40, 50, 20, 95 B. 15, 40, 60, 20
C. 15, 20, 40, 45 D. 45, 40, 15, 20
12. 一组纪录的关键字序列为(33, 66, 43, 25, 27, 71)，为排成非递减序列利用堆排序的方法建立的初始堆为 _____。
A. (66, 33, 43, 25, 27, 71) B. (71, 66, 43, 33, 27, 25)
C. (71, 66, 43, 25, 27, 33) D. (71, 43, 66, 27, 33, 25)
13. 下列说法错误的是 _____。
A. 求最短路径的 Dijkstra 算法中边的权值不可以为负
B. Dijkstra 算法允许图中有回路
C. Floyd 算法中边的权值不可以为负
D. 深度优先搜索算法可用于判断有向图中是否存在回路
14. 某二叉树的前序遍历序列和后序遍历序列正好相反，则该二叉树一定是 _____ 的二叉树。
A. 空或只有一个结点 B. 叶子结点数至多为 1
C. 任一结点无左儿子 D. 任一结点无右儿子
15. 若已知一个栈的入栈序列是 1, 2, 3, … n，其输出序列为 p1, p2, p3, … pn，若 p1 = n，则 pi 为： _____。
A. i B. n-i C. n-i+1 D. 不确定
16. 表达式 $a * (b + c) - d$ 的前缀表达式是 _____。
A. abcdd+- B. -*a+bcd C. abc*+d- D. -+*abcd

二、程序填充题（每格 1.5 分，共 24 分）

1. 已知二叉查找树采用二叉链表存储。下面程序的功能为：删除结点值是 x 的结点，请在下列空格中填写适当语句。要求删除该结点后，此树仍然是一棵二叉排序树，并且高度没有增长（注：可不考虑被删除的结点是根的情况）。下面代码首先查找结点值为 x 的结点。查找成功后，若该结点无左子树，则可直接将其右子树的根结点接到其父亲结点上；若该结点有左子树，则将其左子树中按中序遍历的最后一个结点代替该结点，从而不增加树的高度。

```

template <typename Type>
struct BinaryNode {Type data; BinaryNode *left; BinaryNode *right; ...};
template <typename Type>
struct BinarySearchTree {BinaryNode <Type> *root; ...};
template <typename Type>
void Delete(BinarySearchTree <Type> bst, Type x) {
    BinaryNode <Type> *f, *q, *s, *p = bst.root;
    while (p && p->data != x)
        if ( _____ ) {f = p; p = p->left;}
        else {f = p; p = p->right;}
    if (p == NULL) {cout << "值为 X 的结点不存在\n"; exit(0);}
    if (!p->left) {
        if (f->left == p) _____;
        else f->right = p->right;
        _____;
    }
    else { //被删结点有左子树
        q = p; s = p->left;
        while ( _____ ) {q = s; s = s->right;}
        p->data = s->data;
        if ( _____ ) p->left = s->left;
        else q->right = s->left;
        _____;
    }
}

```

2. 如下为二分查找的非递归算法，试将其填写完整。

```

Template <class ElemtType, class keyType>
int Binsch(ElemtType A[], int n, KeyType K) {
    int low = 0, high = n-1;
    while (low <= high) {
        int mid = _____;
        if (K == A[mid].key) return mid; //查找成功，返回元素的下标
        else
            if (K < [mid].key) _____; //在左子表上继续查找
            else _____; //在右子表上继续查找
    }
    return -1; //查找失败，返回-1
}

```

3. 下面程序段的功能实现数据 x 进栈，要求在下划线处填上正确的语句。

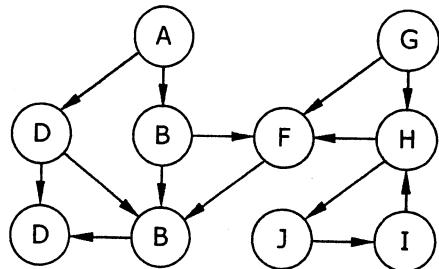
```
struct sqstack {int s[100]; int top; sqstack() {top = _____;}};  
void push(sqstack &stack, int x) {  
    if ( _____ ) cout << "overflow";  
    else {_____ ; _____ }  
}
```

4. 以下程序的功能是将单链表除头节点之外的其余节点逆序连接，请填空。

```
template <typename E> class List {public: E element; Link *next; ...}  
template <typename E> class LList {  
private:  
    Link <E>* head; ...  
public: ...  
void reverse() {  
    Link <E> *p, *q;  
    p = head -> next;  
    if ( _____ ) return;  
    head -> next = NULL;  
    while ( _____ ) {  
        q = p;  
        p = p -> next;  
        q -> next = head -> next;  
        head -> next = _____ ;  
    }  
}
```

三、简答题（每题 8 分，共 24 分）

1. 按照字母从小到大的顺序（从 A 开始），写出该有向图的深度优先搜索和广度优先搜索序列结果，并分别画出所得到的生成树/森林。



2. 某天，一个考古学家发现某种神秘古代语言。这种语言的文字由字母组成，且有且仅有以下十个字母：! @ # \$ % ^ & * ()。经过统计，在某段由该文字书写的文献中，它们分别出现了以下次数：13, 10, 45, 789, 8, 80, 33, 200, 28, 99。画出一棵用以设计这种文字的哈夫曼编码方案的哈夫曼树。

3. 已知一个散列表如下图所示:

		35		20			33		48			59
0	1	2	3	4	5	6	7	8	9	10	11	12

其散列函数为 $h(key) = key \% 13$, 处理冲突的方法为双重散列法, 探查序列为:

$$h_i = (h(key) + i * h_1(key)) \% 13 \quad i = 0, 1, \dots, m-1$$

其中, $h_1(key) = key \% 11 + 1$. 请回答下列问题:

1) 对表中关键字 35, 20, 33 和 48 进行查找时, 所需进行的比较次数各为多少?

2) 该散列表在等概率查找时查找成功的平均查找长度为多少?

四、分析题（每题 8 分，共 16 分）

1. 下列程序是某种常用算法：

```
template <class T>
void A(T a[], int x, int y) {
    int w;
    if (x >= y) return;
    w = C(a, x, y);
    A(a, x, w - 1);
    A(a, w + 1, y);
}

template <class T>
void B(T a[], int arraySIZE) {A(a, 0, arraySIZE - 1);}

template <class T>
int C(T a[], int x, int y) {
    T tmp = a[x];
    do {
        while (x < y && a[y] >= tmp) --y;
        if (x < y) {a[x] = a[y]; _____;}
        while (x < y && a[x] <= tmp) ++x;
        if (x < y) {a[y] = a[x]; _____;}
    } while (x != y);
    a[x] = tmp;
    return x;
}
```

阅读上述程序并回答下列问题：

- 1) 若在主程序中调用函数 B，这样实现了什么功能？
- 2) 请将 C 函数中的空白处填上适当的代码。
- 3) 该算法在最坏情况、最好情况下的时间复杂度分别是多少？

2. 已知下列程序：

```
template <typename E> class LinkList {  
Public:  
    E element;  
    LinkList *next;  
    ...  
}  
LinkList <E> mynote(LinkList <E> L) {           //L是不带头结点的单链表的头指针  
    if (L && L->next) {  
        q = L; L = L->next; p = L;  
        S1:   while (p->next) p = p->next;  
        S2:   p->next = q; q->next = NULL;  
    }  
    return L;  
}
```

请回答下列问题：

- 1) 说明语句 S1 的功能；
- 2) 说明语句 S2 的功能；
- 3) 设链表表示的线性表为 (a_1, a_2, \dots, a_n) ，写出算法执行后的返回值所表示的线性表。

五、设计题（12分）

设计一算法，在一个规模为 N 的无序数组中找出第 k 个大的元素，要求时间复杂度为 $O(N \log_2 k)$ 。
首先用文字描述设计思路，然后写出伪代码。

六、附加题 (10 分)

给下面的以邻接表实现的有向图类增加一个成员函数，要求采用深度优先搜索算法，判断有向图中是否存在顶点 v_i 到顶点 v_j ($i \neq j$) 的路径。要求以非递归方法实现。

bool path_i_j(TypeOfVer vi, TypeOfVer vj);

如果顶点 v_i 和 v_j 之间存在路径，则返回 true；否则返回 false。

```
template <typename TypeOfVer, typename TypeOfEdge>
class adjListGraph {
public:
    bool path_i_j(TypeOfVer vi, TypeOfVer vj);
private:
    struct edgeNode {                                     //邻接表中存储边的结点类
        int end;                                         //终点编号
        edgeNode *next;
        edgeNode (int e, edgeNode *nxt = NULL) {
            end = e; next = nxt;
        }
    };
    struct verNode {                                     //顶点值
        TypeOfVer ver;                                 //顶点值
        edgeNode *head;                                //对应的边表的头指针
        verNode(edgeNode *h = NULL) {head = h;}
    };
    int Vers;                                         //顶点数量
    int Edges;                                        //边的数量
    verNode *verList;                                 //存储顶点的数组的首地址
};
```