

华中师范大学 2010-2011 学年第 2 学期
期末考试试卷 (B 卷)

课程名称 数据结构 课程编号 84810007 任课教师 魏开平、王敬华、陈曙

题型	单选题	填空题	简答题	阅读题	编程题	总分
分值	20	20	20	20	20	100
得分						

得分	评阅人

一、选择题 (每小题 1 分, 共 20 题)

- 下面程序段的时间复杂度为()。

```
for(int i=0; i<m; i++)
    for (int j=0; j<n; j++) a[i][j] = i*j;
```

A. $O(m^2)$ B. $O(n^2)$ C. $O(m*n)$ D. $O(m+n)$
- 在表长为 n 的顺序表上做插入运算, 平均要移动的结点数是()。

A. n B. $n/2$ C. $n/3$ D. $n/4$
- 双链表中某点(已知某地址)前, 插入一新结点, 其所需时间为()。

A. $O(1)$ B. $O(\lg n)$ C. $O(n)$ D. $O(n^2)$
- 对任何一棵二叉树 T , 如果其叶子结点度数为 n_0 , 度为 2 的结点的度数为 n_2 , 则()。

A. $n_0 = n_2 + 1$ B. $n_2 = n_0 + 1$ C. $n_0 = 2n_2 + 1$ D. $n_2 = 2n_0 + 1$
- 有 m 个叶结点的哈夫曼树所具有结点数为()。

A. m B. $m+1$ C. $2m$ D. $2m-1$
- 有 n 个结点的无向图的边数最多为()。

A. $n+1$ B. $n(n-1)/2$ C. $n(n+1)$ D. $2n(n+1)$
- 广度优先遍历类似于二叉树的()。

A. 先序遍历 B. 中序遍历 C. 后序遍历 D. 层次遍历
- 冒泡排序的时间复杂度是()。

A. $O(n)$ B. $O(n \log_2 n)$ C. $O(n^2)$ D. $O(\log_2 n)$
- 快速排序最易发挥其长处的情況是()。

A. 被排序的数据中含有多个相同排序码
 B. 被排序的数据已基本有序
 C. 被排序的数据完全无序
 D. 被排序的数据中的最大值和最小值相差悬殊
- 利用双向链表作线性表的存储结构的优点是()。

A. 便于单向进行插入和删除的操作 B. 便于双向进行插入和删除的操作
 C. 节省空间 D. 便于销毁结构释放空间

11. 设有一个递归算法如下

```

int X(int n) {
    if(n <= 3) return 1;
    else return X(n-2)+X(n-4)+1;
}

```

试问计算 $X(X(5))$ 时需要调用()次 X 函数。

- A. 2 B. 3 C. 4 D. 5
12. 一棵具有 35 个结点的完全二叉树的高度为()。假定空树的高度为-1。
 A. 5 B. 6 C. 7 D. 8
13. 向具有 n 个结点的堆中插入一个新元素的时间复杂度为()。
 A. $O(1)$ B. $O(n)$ C. $O(\log_2 n)$ D. $O(n \log_2 n)$
14. 一个有 n 个顶点和 n 条边的无向连通图一定是()。
 A. 连通的 B. 不连通的 C. 无环的 D. 有环的
15. 在用 Kruskal 算法求解带权连通图的最小(代价)生成树时, 通常采用一个()辅助结构, 判断两个端点是否在同一连通分量上。
 A. 位向量 B. 堆 C. 并查集 D. 生成树顶点集合
16. 给定有 n 个元素的向量, 建立一个有序单链表的时间复杂度是()。
 A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n \log_2 n)$
17. 在一棵具有 n 个结点的二叉树的第 i 层上(假定根结点为第 0 层, i 大于等于 0 而小于等于树的深度)多具有()个结点。
 A. 2^i B. 2^{i+1} C. 2^{i-1} D. 2^n
18. 利用 3, 6, 8, 12 这四个值作为叶子结点的权, 生成一棵霍夫曼树, 该树的带权路径长度为()。
 A. 55 B. 29 C. 58 D. 38
19. 对于具有 e 条边的无向图, 它的邻接表中有()个边结点。
 A. $e-1$ B. e C. $2(e-1)$ D. $2e$
20. 一个对象序列的排序码为 {46, 79, 56, 38, 40, 84}, 采用快速排序(以位于最左位置的对象为基准)的第一次划分结果为()。
 A. {38, 46, 79, 56, 40, 84} B. {38, 79, 56, 46, 40, 84}
 C. {40, 38, 46, 79, 56, 84} D. {38, 46, 56, 79, 40, 84}

得分	评阅人

二、填空题。(每小题 2 分, 共 20 分)

21. 在双链表中, 存储一个结点有三个域, 一个是数据域; 另两个是指针域, 分别指向_____。
22. 设广义表 $C = ((x, (a, b)), y)$, 则 C 的长度为_____, 深度为_____。
23. 森林的后序遍历序列是相应的二叉树的_____遍历序列, 森林的先序遍历序列与相应_____遍历序列相同。
24. 在一棵具有 n 个结点的完全二叉树中, 从树根起, 自上而下、自左而右地给所有结点编号, 设编号为 1。若编号为 i 的结点, 有右孩子, 那么其右孩子的编号为_____; 若编号为 i 的结点是叶子结点, 那么其父结点的编号为_____。

25. 有一稠密图 G , 则 G 采用_____存储较省空间。设有一稀疏图 G , 则采用_____存储较省空间。
26. 在一个链式队列中, 若队头指针与队尾指针的值相同, 则表示该队列至多有_____个结点。
27. 若将一棵树 $A(B(C,D,E), F(G(H,I)))$ 按照左子女-右兄弟表示法转换为二叉树, 该二叉树中度为 2 的结点的个数为_____个。
28. 在一个堆的顺序存储中, 若一个元素的下标为 $i(0 \leq i \leq (n-1)/2)$, 则它的左孩子元素的下标为_____。
29. $n(n>0)$ 个顶点的有向图中顶点的入度的最大值不超过_____。
30. 每次使两个相邻的有序表合并成一个有序表, 这种排序方法叫做_____排序。

得分	评阅人

三. 简答题。(每小题 5 分, 共 20 分)

31. 假定无向图 G 有 6 个结点和 9 条边, 并依次输入这 9 条边为 $(0,1), (0,2), (0,4), (0,5), (1,2), (1,3), (2,3), (3,4), (4,5)$ 。
- (1) 画出 G 的邻接邻接表。
- (2) 根据邻接表从顶点 3 出发, 写出按深度优先搜索法进行遍历的结点序列。

32. 已知一棵二叉树的静态数组表示 (即顺序存储) 如下, 其中 -1 表示空, 请分别写出该二叉树的先序、中序、后序遍历序列。

0	1	2	3	4	5	6	7	8	9	10	11	12
20	8	46	5	15	30	-1	-1	-1	10	18	-1	35

33. 已知一个带权图的顶点集 V 和边集 G 分别为:

$$V = \{0, 1, 2, 3, 4, 5\};$$

$$E = \{(0,1)19, (0,2)21, (0,3)14, (1,2)16, (1,5)5, (2,3)26, (2,4)11, (3,4)18, (4,5)6\};$$

试根据克鲁斯卡尔算法求出最小生成树。

34. 已知一个数据表为{ 48, 25, 56, 32, 40 }, 以第一个元素作为基准, 请写出在进行快速排序的过程中划分后数据表的变化。

得分	评阅人

四、阅读程序写结果 (每小题 5 分, 共 20 分)

35. 在单链表的开始点之前插入一新结点。

```

struct LinkNode {
    int data;
    LinkNode *next;
};
void Deamo( LinkNode *head, int x )
{
    /*head 为不带头结点的单链表
    ListNode *p;
    p = new LinkNode;
    p->data = x;
    _____ (1) _____;
    _____ (2) _____;
}

```

36. 下列函数是在无向图的邻接表中删除一条边的算法。

```

void deledege(ALGRAPH *G, int i, int j)
{
    edgenode *p, *q;
    p = G->adjlist[i].firstedge;
    if ( p->adjvex == j ) { G->adjlist[i].firstedge = p->next; free(p); }
    else {
        while ( p->next->adjvex != j && p->next )
            _____ (1) _____;
        if ( p->next != null ) { q = p->next; _____ (2) _____; free(q); }
    }
    p = G->adjlist[i].firstedge;
    if ( p->adjvex == i ) { G->adjlist[i].Firstedge = p->next; free(p); }
    else {
        while ( p->next->adjvex != i && p->next )
            _____ (3) _____;
        if ( p->next != null ) { q = p->next; _____ (4) _____; free(q); }
    }
}
}

```


37. 已知奇偶转换排序如下所述：第一趟对所有奇数的 i ，将 $a[i]$ 和 $a[i+1]$ 进行比较，第二趟对所有偶数的 i ，将 $a[i]$ 和 $a[i+1]$ 进行比较，每比较时若 $a[i] > a[i+1]$ ，则将两者交换，以后重复上述两趟过程交换进行，直至整个数组有序请完善下列程序。

```
void oesort ( int a[ n ] )
{
    int i, flag;
    do {
        flag = 0;
        for ( i = 1; i < n; i += 2, i++ ) /* 奇数扫描 */
            if ( a[i] > a[i+1] ) {
                _____(1)_____;
                t = a[i+1]; a[i+1] = a[i]; a[i] = t;
            }
        for ( _____(2)_____ /* 偶数扫描 */
            if ( _____(3)_____ ) {
                flag = 1; t = a[i+1]; a[i+1] = a[i]; a[i] = t;
            }
    } while ( _____(4)_____ );
}
```

38. $rear$ 是指向以循环链表表示的队列的队尾指针，EnQueue 函数实现把 x 插入到队尾的操作。DeQueue 函数实现删除队头元素并赋给 x 的操作。根据题意按标号把合适的内容填写到算法后面相应标号的位置。

```
void EnQueue( ListNode *rear, int x )
{
    ListNode *p;
    p = new ListNode;
    p->data = x;
    p->next = NULL;
    _____(1)_____;
    rear = p;
};

bool DeQueue( ListNode *rear, int &x )
{
    if ( rear == NULL ) return false;
    if ( rear->next == rear ) { x = rear->data; delete rear; rear = NULL; return false; }
    ListNode *p = rear->next;
    rear->next = p->next;
    _____(2)_____;
    _____(3)_____;
    return true;
}
```

得分	评阅人

五、编写程序（函数）（每小题 10 份，共 20 分）

39. 设有一个线性表 $(e_0, e_1, \dots, e_{n-2}, e_{n-1})$ 已存放在一个一维数组 $A[Size]$ 中的前 n 个数组元素位置上。请编写一个函数将这个线性表原地逆置，即将数组的前 n 个元素内容置换为 $(e_{n-1}, e_{n-2}, \dots, e_1, e_0)$ 。函数的原型为：
- ```
void inverse(int A[], int n);
```

40. 已知二叉树中的结点类型用  $BinTreeNode$  表示，被定义为：

```
struct BinTreeNode {
 char data;
 BinTreeNode *left, *right;
};
```

其中  $data$  为结点值域， $left$  和  $right$  分别为指向左、右孩子结点的指针域，根据下面函数声明编写出交换一棵二叉树中所有结点的左、右指针域值的算法，算法中参数  $BT$  初始指向这棵二叉树的根结点。函数的原型为：

```
void BTreeSwap(BinTreeNode* BT);
```



华中师范大学 2010-2011 学年第 2 学期

期末考试试卷 (B 卷) - 参考答案

课程名称 数据结构 课程编号 84810007 任课教师 魏开平、王敬华、陈曙

一、选择题 (每小题 1 分, 共 20 题)

CBAADBCCBCACDCCCADC

二、填空题。(每小题 2 分, 共 20 分)

21. 前驱 后继
22. 23
23. 中序 先序
24.  $2i+1$   $i/2$
25. 邻接矩阵 邻接表
26. 1
27. 2
28.  $2i$
29.  $n-1$
30. 归并

三、简答题。(每小题 5 分, 共 20 分)

31. 0: 1->2->->4

1: 0->2->3

2: 0->1->3

3: 1->2->4

4: 3->5

5: 0->4

深度遍历序列: 3->1->0->2->4->5

32. 先序: 20 8 5 15 10 18 46 30 35

中序: 5 8 10 15 18 20 30 35 46

后序: 5 10 18 15 8 35 30 46 20

33. (1, 5), (4, 5), (2, 4), (3, 4), (0, 3)

34. 41 25 32 48 56

32 25 41 48 56

23 32 41 48 56

四、阅读程序写结果 (每小题 5 分, 共 20 分)

35. `p->next = head; head = p;`

36. `p = p->next; p->next = q->next; p = p->next; p->next = q->next;`

37. `flag = 1; i=0; i<n; i++, i++; a[i]>a[i+1]`

38. `rear->next = p; x = p->data; delete p;`

五、编写程序 (函数) (每小题 10 份, 共 20 分)

39. 略

40. 略