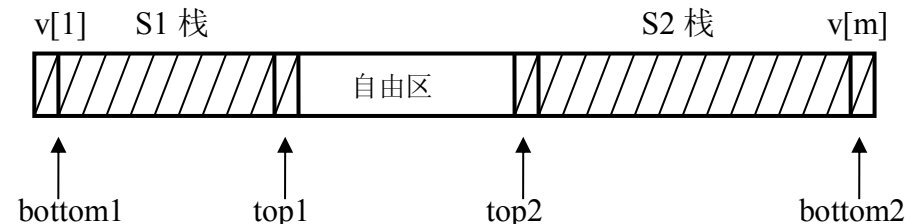


试卷 6

一、单选题 (本大题共 20 小题。每小题 1 分，共 20 分。)

- 数据结构研究的内容不涉及 ()。
 - 数据如何组织
 - 数据如何存储
 - 数据的运算如何实现
 - 算法用什么语言来描述
- 算法分析的目的是 ()。
 - 找出数据结构的合理性
 - 研究算法中输入和输出关系
 - 分析算法的效率以求改进
 - 分析算法的易懂性
- 在线性表的顺序存储结构中，已知首元的存储地址是 200，每个元素的长度为 2，则第 5 个元素的存储首址是 ()。
 - 210
 - 208
 - 200
 - 220
- 非空的循环单链表 head 的尾结点 *rear 满足 ()。
 - $\text{rear} \rightarrow \text{next} == \text{NULL}$
 - $\text{rear} == \text{NULL}$
 - $\text{rear} \rightarrow \text{next} == \text{head}$
 - $\text{rear} == \text{head}$
- 在一个单链表中，若 *p 不是最后的结点，在 *p 之后插入结点 *s，则应执行语句 ()。
 - $\text{s} \rightarrow \text{next} = \text{p}; \quad \text{p} \rightarrow \text{next} = \text{s};$
 - $\text{s} \rightarrow \text{next} = \text{p} \rightarrow \text{next}; \quad \text{p} \rightarrow \text{next} = \text{s};$
 - $\text{s} \rightarrow \text{next} = \text{p} \rightarrow \text{next}; \quad \text{p} = \text{s};$
 - $\text{p} \rightarrow \text{next} = \text{s}; \quad \text{s} \rightarrow \text{next} = \text{p};$
- 若用大小为 8 的一维数组来实现循环队列，且当前 rear 和 front 的值分别为 2 和 5，当从队列中删除一个元素，再加入两个元素后，rear 和 front 的值分别是 ()。

A. 0 和 4 B. 3 和 6 **C. 4 和 6** D. 6 和 4

- 递归函数 $f(n) = nf(n-1) \quad (n \geq 1)$ 的递归出口是 ()。
 - $f(1) = 0$
 - $f(1) = 1$
 - $f(0) = 1$
 - $f(n) = n$
 - 在下图中，若整个存储区被占满了才溢出，则栈满的条件为 ()。
 - $\text{top2} == \text{top1} - 1$
 - $\text{top2} == \text{top1}$
 - $\text{top1} == \text{top2} - 1$
 - $\text{top1} == \text{top2} + 1$
- 
- 将一个对称矩阵 $A[10][10]$ 的下三角按行优先放入一维数组 $B[100]$ 中，则元素 $A[5][6]$ 在数组 B 中的序号为 ()。
 - 19
 - 26
 - 21
 - 15
 - 已知某二叉树的后序遍历序列是 dabec，中序遍历序列是 debac，则其前序遍历序列是 ()。
 - acbed
 - decab
 - deabc
 - cedba**
 - 在一棵非空二叉树的中序遍历序列中，根结点的右边 ()
 - 只有右子树上的所有结点
 - 只有右子树上的部分结点
 - 只有左子树上的部分结点
 - 只有左子树上的所有结点
 - 在一棵树的“左孩子—右兄弟”表示法中，一个结点的右子女是该结点的 () 结点。
 - 兄弟
 - 堂兄弟
 - 祖先
 - 子孙

13. 以数据集 { 2, 3, 4, 7, 8, 9 } 为结点权值所构造的 huffman 树的带权路径长度为 ()。

- A. 80 B. 81 C. 82 D. 79

14. 具有 4 个顶点的无向完全图有 () 条边。

- A. 6 B. 12 C. 16 D. 20

15. 一个有 n 个顶点的无向图是连通的, 至少应有 () 条边。

- A. n B. $n+1$ C. $n-1$ D. $n/2$

16. 一个图中包含 k 个连通分量, 若按深度优先算法访问所有结点, 则必须调用 () 次深度优先遍历算法。

- A. 1 B. $k-1$ C. k D. $k+1$

17. 关键路径是 AOE 网中, 从源点到汇点 ()。

- A. 最长的路径 B. 最短的路径
C. 最长的回路 D. 最短的回路

18. 顺序查找法适合于存储结构为 () 的线性表。

- A. 散列存储 B. 顺序存储或链式存储
C. 压缩存储 D. 索引存储

19. 采用折半查找法查找长度为 n 的线性表时, 平均查找长度为 ()。

- A. $O(n^2)$ B. $O(n\log_2 n)$ C. $O(n)$ D. $O(\log_2 n)$

20. 下列查找方法中, 原理与其它不同的是 ()。

- A. 顺序查找 B. 折半查找 C. 分块查找 D. 哈希查找

二、判断题 (下面命题你认为正确的在题前的括号内打“√”, 错误的打“×”, 并填入答题卡。本大题共 10 小题, 每小题 1 分, 共 10 分)

1. () 数据结构在计算机中的表示称为数据的物理结构, 又称存储结构。

2. () 在相同规模 n 下, 复杂度 $O(n)$ 的算法在时间上总是优于复杂度 $O(2^n)$ 的算法。

3. () 每种数据结构都有插入、删除和查找三种基本操作。

4. () 线性表的逻辑顺序和存储顺序总是一致的。

5. () 二维数组是数据元素为线性表的线性表。

6. () 顺序队列和循环队列的队满和队空的判断条件是一样的。

7. () 深度为 K 的完全二叉树至少有 2^{k-1} 个结点。

8. () 线索二叉树中, t 结点没有左孩子的充要条件是 $t \rightarrow ltag == 0$ 。

9. () 在排序方法中, 从未排序的序列中一次取出元素与已排序序列中的元素进行比较, 将其放在已排序序列的适当位置, 这种方法叫做插入排序法。

10. () 基数排序是稳定的排序, 一般用于多关键字排序。

三、填空题 (将你认为正确的答案填在相应的空中, 并填入答题卡。本大题共 10 小题, 每空 2 分, 共 20 分)

1. 下面程序段的时间复杂度为_____。

```
for ( i=1; i<= n; i++ )  
    for ( j=1; j<=i; j++ )  
        a [ i ][ j ] = 0;
```

2. 在 n 个结点的单链表中要删除已知结点 $*p$, 需找到它的_____。

3. 在具有 n 个单元的循环队列中, 队满时共有 _____ 个元素。

4. 若已知一个栈的入栈顺序是 1, 2, 3, ..., n, 其输出序列为 p1, p2, p3, ..., pn, 若 p1=n, 则 pi 为_____。
5. 对称矩阵 A 按行序将其下三角存放在一维数组 B [1..n(n+1)/2] 中, 矩阵中的元素 a[i][j](i≥j), 在一维数组 B 中的下标为_____。
6. 一般情况下, 稀疏矩阵的压缩存放有两种方式, 即_____表示法和十字链表表示法。
7. 设一棵完全二叉树有 700 个结点, 则共有 _____ 个叶子结点。
8. 某二叉树的先序遍历和后序遍历的顺序正好相反, 则该二叉树一定是 _____。
9. G 是一个非连通的无向图, 共有 28 条边, 则它至少有_____ 个结点。
10. 在所有的查找方法中, 平均查找长度与结点个数 n 无关的查找方法是_____。

四、综合题 (本大题共 5 小题。每小题 8 分, 共 40 分)

1. 算法填空。在画有横线的地方填写合适的内容。

从类型为 SqList 的线性表 L 中删除与 x 值相等的所有元素。

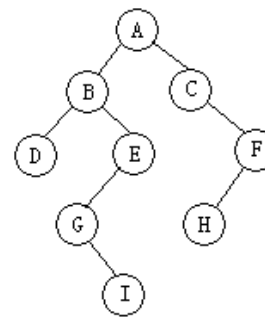
```
void func1 (SqList & L, ElemType x)
{
    int i = _____, j;
    while ( i < L.length)
    {
        if ( L.elem[ i ] == _____ )
        {
            for ( j=i+1; j < L.length; j++)
                L.elem [ j-1] = L.elem [ j];
```

```
                _____;
            }
            else i++;
        }
    }
```

2. 阅读以下算法。并简述算法的功能。

```
int func2 ( LNode* &L)
{
    int max;
    LNode* p = L->next;
    if ( L->next == NULL)
    {
        printf ("Linked list is empty!");
        return FALSE;
    }
    max = p->data;
    while ( p != NULL )
    {
        if ( max < p->data)
            max = p->data;
        p = p->next;
    }
    return max;
}
```

3. 试写出下图二叉树的中序遍历序列, 并加上相应的中序线索。



中序序列为:

4. 根据给出的权值，构造一棵哈夫曼树，并计算其 WPL 值。

$W = \{ 2, 3, 4, 7, 8, 9 \}$

5. 试将下面数据表建成一个小顶堆。

(70, 12, 20, 31, 1, 5, 44, 66, 61, 200, 30, 80, 150, 4, 28)

五、编写算法（本题 10 分）

设计算法，将一个带头结点的单链表 A 分解为两个具有相同结构的链表 B 和 C。其中 B 表中的结点为 A 表中值为奇数的结点，而 C 表中的结点为 A 表中值为偶数的结点。（要求利用原表的结点。提示：设 $\text{odd}(x)$ 为判奇数函数， x 为奇数，返回 1，否则返回 0。该函数可用。）

单链表的类型定义如下：

```
typedef struct node{
    DataType    data;
    struct node *next;
}LinkNode, *LinkList;
```

算法的函数原型给定为：

```
void func3 ( linklist &A, linklist &B, linklist &C )
```

试卷 6 答案和评分标准

一、单选题（每小题 1 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	D	C	B	C	B	C	C	C	B	D
题号	11	12	13	14	15	16	17	18	19	20
答案	A	A	A	A	C	C	A	B	D	D

二、判断题（每小题 1 分，共 10 分）

题号	1	2	3	4	5	6	7	8	9	10
答案	✓	✓	✓	✗	✓	✗	✓	✗	✓	✓

三、填空题（每小题 2 分，共 20 分）

题号	1.	2.	3.	4.
答案	$O(n^2)$	直接前驱	$n-1$	$n-i+1$
题号	5.	6.	7.	8.
答案	$i*(i+1)/2 + j - 1$	三元组表	350	高度等于其结点数
题号	9.	10.		
答案	9	哈希查找法		

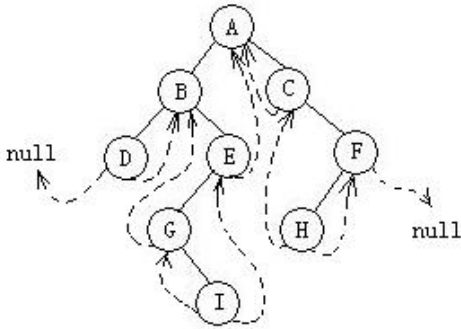
四、综合题（每小题 8 分，共 40 分）

1. 0 -----2 分

x -----3 分
L.length-- -----3 分

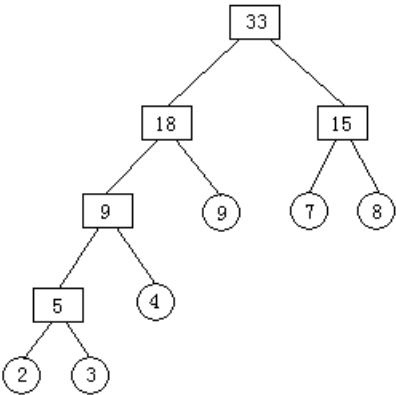
2. 算法功能为：
在带头结点的单链表 L 中，查找具有最大值的结点。（4 分）
该最大值由函数返回。（2 分）
若单链表为空，则终止运行。（2 分）

3. 中序序列：DBGIEACHF -----4 分（酌情给分）



-----4 分（酌情给分）

4. 构造的哈夫曼树为下图所示。



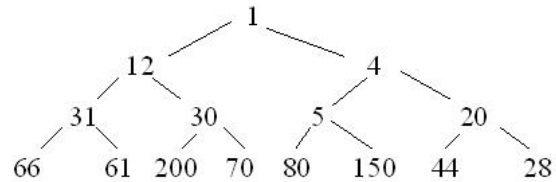
-----4 分（酌情给分）

$$WPL = (9+8+7) * 2 + 4 * 3 + (2+3) * 4 = 80$$

———4分(酌情给分)

5. 数据表 (70, 12, 20, 31, 1, 5, 44, 66, 61, 200, 30, 80, 150, 4, 28)

对应的小顶堆为:



每一层对, 给2分(酌情给分)。

五、编写算法 (本题 10 分)

```
void func ( linklist &A, linklist &B, linklist &C )
```

```
{
```

```
    linklist p, q, rc;
```

```
    C = ( linklist *) malloc(sizeof( node ));
```

```
    C->next = NULL;
```

```
    rc = C;
```

```
    p = A->next; q = A;
```

```
    while ( p && A )
```

```
    {
```

```
        if ( odd ( p->data ) )
```

```
        {
```

```
            q = p;
```

```
            p = p->next;
```

```
        }
```

```
    else
```

```
    {
```

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

```
        rc->next = p;
```

```
        rc = p;
```

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

(初始化 2分)

(循环 2分)

(奇数处理 2分)

(偶数处理 2分)

```
    }
} //endwhile
rc->next = NULL;
B = A;
}
```

(指针处理 2分)