

云南省普通高校“专升本”招生考试

数据结构 机考模拟试题

数据结构（共 4 大题，54 小题，150 分）

第一大题、判断对错题（判断题，共 10 小题，每小题 1 分，共 10 分）

- 1、数据结构是一门研究数值计算的程序设计问题中计算机的操作对象以及他们之间的关系和操作等等的学科。
- 2、数据元素是数据的基本单位，在计算机程序中通常作为一个整体进行考虑和处理。
- 3、线性结构的集合中每个数据元素都只有一个前驱。
- 4、栈是限定只能在表的一端进行插入和删除的线性表。
- 5、队列是限定只能在表的一端进行插入和删除的线性表。
- 6、非空的双向循环链表中任何结点的前驱指针均不为空。
- 7、通过一系列的过程调用语句间接或直接调用自己的过程称做递归过程。
- 8、在以链表存储串值时，定义串的存储密度=串值所占的存储位/实际分配的存储位。那么，存储密度越小，操作越不方便。
- 9、广义表是一种递归定义的数据结构，因此，它也是一种线性结构。
- 10、快速排序是由冒泡排序改进而得的，使 n 个记录排序的时间复杂度由 $O(n^2)$ 降至 $O(n\log n)$ 。

第二大题、单项选择题（单选题，共 20 小题，每小题 2 分，共 40 分）

- 1、队列是一种（ ）的线性表。
A、先进先出
B、先进后出
C、只能插入
D、只能删除
- 2、设指针变量 top 指向当前链式栈的栈顶，则删除栈顶元素的操作序列为（ ）。
A、 $top=top+1;$
B、 $top=top-1;$
C、 $top->next=top;$
D、 $top=top->next;$
- 3、设指针变量 p 指向单链表中结点 A ，指针变量 s 指向被插入的结点 X ，则在结点 A 的后面插入结点 X 需要执行的语句序列： $s->next=p->next;$ （ ）；。
A、 $p->next=s->next$
B、 $p->next=s$
C、 $s=p->next$
D、 $s->next = p$
- 4、设顺序线性表中有 n 个数据元素，则删除表中第 i 个元素需要移动（ ）个元素。
A、 i
B、 $n+1-i$
C、 $n-1-i$
D、 $n-i$
- 5、现有字符串变量 x ，并且 $x="12345789"$ ，那么串 x 的长度是（ ）。
A、13

- B、11
C、9
D、8
- 6、设某棵二叉树中有 2001 个结点，则该二叉树的最小高度为（ ）。
A、9
B、10
C、11
D、12
- 7、二叉树的第 k 层的结点数最多为（ ）。
A、 2^k-1
B、 $2K+1$
C、 $2K-1$
D、 2^{k-1}
- 8、设某棵二叉树的中序遍历序列为 ABCD，先序遍历序列为 CABD，则后序遍历该二叉树得到序列为（ ）。
A、BADC
B、BCDA
C、CDAB
D、CBDA
- 9、设某无向图有 n 个顶点，则该无向图的邻接表中有（ ）个表头结点。
A、2n
B、n
C、n/2
D、n(n-1)
- 10、设有 6 个结点的无向图，该图至少应有（ ）条边才能确保是一个连通图。
A、5
B、6
C、7
D、8
- 11、设无向图 G 中有 n 个顶点 e 条边，则其对应的邻接表中的表头结点和表结点的个数分别为（ ）。
A、n, e
B、e, n
C、2n, e
D、n, 2e
- 12、设某有向图的邻接表中有 n 个表头结点和 m 个表结点，则该图中有（ ）条有向边。
A、n
B、n-1
C、m
D、m-1
- 13、顺序查找不论在顺序线性表中还是在链式线性表中的时间复杂度为（ ）。
A、 $O(n)$
B、 $O(n^2)$
C、 $O(n^{1/2})$

D、 $O(\log_2 n)$

14、设一个顺序有序表 $A[1:14]$ 中有 14 个元素，则采用二分法查找元素 $A[4]$ 的过程中比较元素的顺序为 ()。

- A、 $A[1], A[2], A[3], A[4]$
- B、 $A[7], A[6], A[5], A[4]$
- C、 $A[7], A[3], A[5], A[4]$
- D、 $A[7], A[5], A[3], A[4]$

15、二叉排序树中左子树上所有结点的值均 () 根结点的值。

- A、小于
- B、大于
- C、等于
- D、不等于

16、设一组初始记录关键字序列为(Q, H, C, Y, P, A, M, S, R, D, F, X)，则按字母升序的第一趟冒泡排序结束后的结果是 ()。

- A、F, H, C, D, P, A, M, Q, R, S, Y, X
- B、P, A, C, S, Q, D, F, X, R, H, M, Y
- C、A, D, C, R, F, Q, M, S, Y, P, H, X
- D、H, C, Q, P, A, M, S, R, D, F, X, Y

17、

设有向无环图 G 中的有向边集合 $E = \{ \langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 4 \rangle, \langle 1, 4 \rangle \}$ ，则下列属于该有向图 G 的一种拓扑排序序列的是 ()。

- A、1, 2, 3, 4
- B、2, 3, 4, 1
- C、1, 4, 2, 3
- D、1, 2, 4, 3

18、设某哈夫曼树中有 199 个结点，则该哈夫曼树中有 () 个叶子结点。

- A、99
- B、100
- C、101
- D、102

19、设一组初始记录关键字序列为(60, 80, 55, 40, 42, 85)，则以第一个关键字 45 为基准而得到的一趟快速排序结果是 ()。

- A、40, 42, 60, 55, 80, 85
- B、42, 45, 55, 60, 85, 80
- C、42, 40, 55, 60, 80, 85
- D、42, 40, 60, 85, 55, 80

20、程序段 $s=i=0; do \{i=i+1; s=s+i; \} while(i \leq n);$ 的时间复杂度为 ()。

- A、 $O(n)$
- B、 $O(n \log_2 n)$
- C、 $O(n^2)$
- D、 $O(n^3/2)$

第三大题、多项选择题（多选题，共 10 小题，每小题 3 分，共 30 分）

- 1、通常设计一个“好”的算法考虑达到的目标不包括
 - A、正确性
 - B、可行性
 - C、可读性
 - D、健壮性
 - E、唯一性
- 2、以下数据结构中哪些是线性结构？（ ）
 - A、队列
 - B、栈
 - C、线性表
 - D、二叉树
- 3、下面关于线性表的叙述正确的是（ ）。
 - A、线性表采用顺序存储必须占用一片连续的存储空间
 - B、线性表采用链式存储不必占用一片连续的存储空间
 - C、线性表采用链式存储便于插入和删除操作的实现
 - D、线性表采用顺序存储便于插入和删除操作的实现
- 4、两个字符串相等的充要条件是（ ）。
 - A、除空格以后，两个字符串的长度相等
 - B、两个字符串中对应位置上的字符相等
 - C、两个字符串的长度相等
 - D、两个字符串都不可以为空字符串
- 5、根据 e 与 $n \log n$ 的关系，图可以分为（ ）图和（ ）图。
 - A、有限图
 - B、有向图
 - C、无限图
 - D、无向图
 - E、稀疏图
 - F、稠密图
- 6、以下算法属于稳定的排序算法有（ ）。
 - A、冒泡排序
 - B、交换排序
 - C、选择排序
 - D、插入排序
 - E、快速排序
 - F、归并排序
- 7、时间复杂度受数据初始状态影响的排序算法是（ ）。
 - A、堆排序
 - B、冒泡排序
 - C、希尔排序
 - D、快速排序
- 8、以下哪些算法的最坏情况时间复杂度为 $O(n^2)$ 。（ ）
 - A、堆排序
 - B、冒泡排序
 - C、插入排序

- D、快速排序
 E、归并排序
 9、以下哪些算法的空间复杂度为 $O(1)$ 。()
 A、堆排序
 B、冒泡排序
 C、插入排序
 D、快速排序
 E、归并排序
 10、() 二叉排序树不可以得到一个从小到大的有序序列。
 A、先序遍历
 B、中序遍历
 C、后序遍历
 D、层次遍历

第四大题、阅读应用题（单选题，共 14 小题，每小题 5 分，共 70 分）

1、

- 1) 阅读代码，根据已有信息，从备选答案上选择最适合的答案。↵

```
//线性表的双向链表存储结构↵
typedef struct DuLNode{↵
    ElemType data;↵
    struct DuLNode *prior;↵
    struct DuLNode *next;↵
}DuLNode, * DuLinkList;↵
↵
Void ListInsert_DuL(DuLinkList &L, DuNode *p, DuNode *s)↵
//在带头结点的双循环链表 L 中 p 结点前插入 s 结点↵
    _____ 1 _____;↵
    p->prior->next=s;↵
    s->next=p;↵
    _____ 2 _____;↵
} // ListInsert_DuL{↵
↵
Void ListDelete_DuL(DuLinkList &L, DuNode *p, ElemType &e)↵
//删除带头结点的双向循环链表 L 中 p 结点，并以 e 返回它的数据元素↵
    e=p->data;↵
    _____ 3 _____;↵
    p->next->prior=p->prior;↵
    _____ 4 _____;↵
}↵
```

- (1)、根据已有信息，从备选答案上选择最适合的答案。

- A、`p->prior=p->next`
 B、`p->prior=p->next`

- C、p->prior=p->next
- D、p->prior=p->next
- E、p->prior=s
- F、s->prior=p->prior
- G、p->next=p
- H、delete e;
- I、delete p;
- J、delete &p;

(2)、根据已有信息，从备选答案上选择最适合的答案。

- A、s=p->next
- B、s->prior->prior=p->prior
- C、P->prior->next=p->next
- D、p->prior=p->next
- E、p->prior=s
- F、s->prior=p->prior
- G、p->next=p
- H、delete e;
- I、delete p;
- J、delete &p;

(3)、根据已有信息，从备选答案上选择最适合的答案。

- A、s=p->next
- B、s->prior->prior=p->prior
- C、P->prior->next=p->next
- D、p->prior=p->next
- E、p->prior=s
- F、s->prior=p->prior
- G、p->next=p
- H、delete e;
- I、delete p;
- J、delete &p;

(4)、根据已有信息，从备选答案上选择最适合的答案。

- A、s=p->next
- B、s->prior->prior=p->prior
- C、P->prior->next=p->next
- D、p->prior=p->next
- E、p->prior=s
- F、s->prior=p->prior
- G、p->next=p
- H、delete e;
- I、delete p;
- J、delete &p;

2、

2) 阅读代码，根据已有信息，从备选答案上选择最适合的答案。↵

//广义表的存储表示↵

```
typedef enum{ATOM=0,list=1}ElemTag;//分别标记原子和子表↵
```

↵

```
typedef struct GLNode{↵
```

```
ElemTag tag;↵
```

```
union{↵
```

```
AtomType data;//data 是原子结点的值域，AtomType 由用户定义↵
```

```
Struct{struct GLNode *hp, * tp;}ptr;↵
```

```
//ptr 是表结点的指针域，ptr.hp 和 ptr.tp 分别指向表头和表尾↵
```

```
};↵
```

```
*GList;//广义表类型↵
```

```
Void OutputGList(GList LS)↵
```

```
//由广义表存储结构递归打印广义表逻辑结构↵
```

```
if(!LS)cout<<"()";↵
```

```
else{↵
```

```
if(LS->tag==ATOM)cout<<LS->data;↵
```

```
else{↵
```

```
cout<<'(';↵
```

```
p=LS;↵
```

```
while(p){↵
```

```
_____5_____;↵
```

```
p=p->ptr.tp;↵
```

```
if(p) cout<<' ';↵
```

```
}//while↵
```

```
cout<<'(';↵
```

```
}//else↵
```

```
}//else↵
```

```
}//outputGList↵
```

- A、 OutputGList(p)
- B、 OutputGList(ptr.hp)
- C、 OutputGList(ptr.tp)
- D、 OutputGList(p->ptr.hp)
- E、 OutputGList(p->ptr.tp)

3、

3) 阅读代码，根据已有信息，从备选答案上选择最适合的答案。

```
//二叉树的二叉链表存储表示
typedef struct BiTNode{
    TElemType data;
    Struct BiTNode *lchild, *rchild;//左、右孩子指针
}BiTNode, *BiTree;

Void Preorder(BiTree __ 1 __,void(* visit)(BiTree))
//先序遍历以 T 为根指针的二叉树
If(T){
    visit(T);
    _____ 2 _____;
    _____ 3 _____;
}
}
```

(1)、根据已有信息，从备选答案上选择最适合的答案。

- A、*T
- B、T
- C、&T
- D、Preorder(T->rchild,visit)
- E、Preorder(visit,T->rchild)
- F、Preorder(T,visit)
- G、Preorder(T->lchild,visit)
- H、Preorder(visit,T->lchild)

(2)、根据已有信息，从备选答案上选择最适合的答案。

- A、*T
- B、T
- C、&T
- D、Preorder(T->rchild,visit)
- E、Preorder(visit,T->rchild)
- F、Preorder(T,visit)
- G、Preorder(T->lchild,visit)
- H、Preorder(visit,T->lchild)

(3)、根据已有信息，从备选答案上选择最适合的答案。

- A、*T
- B、T
- C、&T
- D、Preorder(T->rchild,visit)
- E、Preorder(visit,T->rchild)
- F、Preorder(T,visit)
- G、Preorder(T->lchild,visit)
- H、Preorder(visit,T->lchild)

4、

4) 阅读代码，根据已有信息，从备选答案上选择最适合的答案。

对所示图 G1，从 v3 出发对 G1 进行广度优先遍历到结束，得到“广度优先”所访问的顶点序列为 $V_3 \rightarrow V_2 \rightarrow V_1 \rightarrow \underline{1} \rightarrow \underline{2} \rightarrow \underline{3} \rightarrow V_9 \rightarrow V_8 \rightarrow V_7$

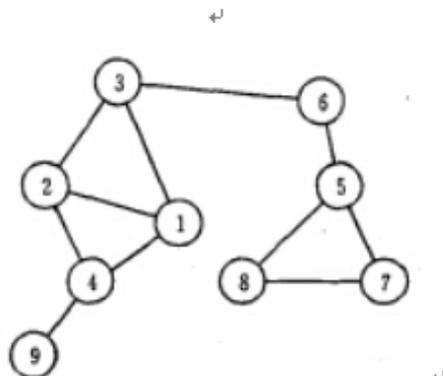


图 G1

(1)、根据已有信息，从备选答案上选择最适合的答案。

- A、V4
- B、V5
- C、V6
- D、V9

(2)、根据已有信息，从备选答案上选择最适合的答案。

- A、V4
- B、V5
- C、V6
- D、V9

(3)、根据已有信息，从备选答案上选择最适合的答案。

- A、V4
- B、V5
- C、V6
- D、V9
- 5、

5) 阅读代码，根据已有信息，从备选答案上选择最适合的答案。

```

typedef struct node
//二叉排序树节点的结构
    ElemType data;//定义数据域
    struct node * lchild;//左孩子指针
    struct node * rchild;//右孩子指针
}BitreeNode;

int SearchBitree(BitreeNode * t ,ElemType k )
//在根据指针 t 所指的二叉排序树上查找数据元素 k 是否存在
//若查找成功，返回 1；否则，返回 0
    _____ 1 _____;
    if(t!=NULL)
    {
        p=t;
        while(p)
        {
            if(p->data==k) return 1;
            else if( _____ 2 _____ ) _____ 3 _____;
            else p=p->rchild;
        }
    }
    Return 0;
}
    
```

(1)、根据已有信息，从备选答案上选择最适合的答案。

- A、 BitreeNode *p
- B、 BitreeNode p
- C、 p->data > k
- D、 p=p->lchild
- E、 p=p->rchild
- F、 p->data<k
- G、 p->data!=k

(2)、根据已有信息，从备选答案上选择最适合的答案。

- A、 BitreeNode *p
- B、 BitreeNode p
- C、 p->data > k
- D、 p=p->lchild
- E、 p=p->rchild
- F、 p->data<k
- G、 p->data!=k

(3)、根据已有信息，从备选答案上选择最适合的答案。

- A、 BitreeNode *p
- B、 BitreeNode p

- C、`p->data > k`
- D、`p=p->lchild`
- E、`p=p->rchild`
- F、`p->data<k`
- G、`p->data!=k`