

2026 年 4 月高等教育自学考试

数据结构试题

课程代码:13181

1. 请考生按规定用笔将所有试题的答案涂、写在答题纸上。

2. 答题前,考生务必将自己的考试课程名称、姓名、准考证号用黑色字迹的签字笔或钢笔填写在答题纸规定的位置上。

选择题部分

注意事项:

每小题选出答案后,用 2B 铅笔把答题纸上对应题目的答案标号涂黑。如需改动,用橡皮擦干净后,再选涂其他答案标号。不能答在试题卷上。

一、单项选择题:本大题共 10 小题,每小题 2 分,共 20 分。在每小题列出的备选项中只有一项是最符合题目要求的,请将其选出。

- 关于算法特性描述正确的是
 - 至少有一个输入值
 - 仅有一个输入值
 - 至少有一个输出值
 - 可以没有输出值
- 设带头结点的单循环链表头指针为 head, 指针变量 p 指向尾结点的条件是
 - $p \rightarrow \text{next} \rightarrow \text{next} == \text{head}$
 - $p \rightarrow \text{next} == \text{head}$
 - $p \rightarrow \text{next} \rightarrow \text{next} == \text{NULL}$
 - $p \rightarrow \text{next} == \text{NULL}$
- 若用数组存放循环队列的元素, 头指针 front 指向队头元素位置, 尾指针 rear 指向队尾元素后面的空位置, 则队列空的判定条件为
 - $(\text{rear}+1) \% n == \text{front}$
 - $(\text{front}+1) \% n == \text{rear}$
 - $\text{rear}+1 == \text{front}$
 - $\text{rear} == \text{front}$
- 在稀疏矩阵的存储结构中, 保存三元组表采用的是
 - 一维数组
 - 二维数组
 - 三维数组
 - 四维数组
- 已知采用链式存储含有 80 个结点的二叉树有 79 个分支, 每个结点都有两个指针域和一个数据域, 则空指针域的个数为
 - 79
 - 80
 - 81
 - 82



6. 关于图的遍历描述正确的是
- A. 图的遍历只用于连通图
 - B. 图的深度优先遍历序列是唯一的
 - C. 图的深度优先遍历是一个非递归过程
 - D. 图的遍历是从某个顶点出发对每一个顶点访问且仅访问一次
7. 含有 n 个顶点的无向连通图的生成树边数是
- A. n
 - B. $2n$
 - C. $n+1$
 - D. $n-1$
8. 不改变关键字相同元素相对位置的排序方法是
- A. 不稳定的
 - B. 稳定的
 - C. 最优的
 - D. 最快的
9. 索引顺序查找又称为分块查找, 该方法要求
- A. 块内有序
 - B. 块间有序
 - C. 各块等长
 - D. 链式存储
10. 哈希方法使用的存储结构是
- A. 数组
 - B. 链表
 - C. 栈
 - D. 队列

非选择题部分

注意事项:

用黑色字迹的签字笔或钢笔将答案写在答题纸上, 不能答在试题卷上。

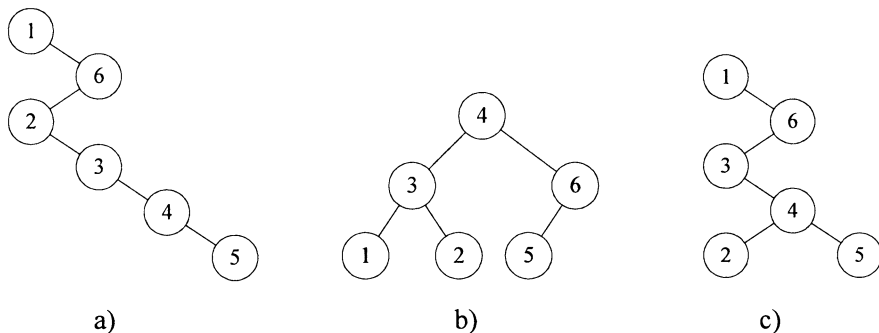
二、填空题: 本大题共 10 空, 每空 2 分, 共 20 分。

11. 根据数据元素的关键字计算该元素的物理存储位置的存储方法是____存储方法。
12. 设线性表 $s=(1,3,5,7,9)$, 执行 $\text{find}(s,3)$ 操作后的返回值为____。
13. 栈操作中, 只要栈不满, 就允许____。
14. 长度为 0 的广义表称为____。
15. 根据先序遍历序列和____遍历序列可以唯一确定一棵二叉树。
16. 二叉树叶结点的带权路径长度定义为叶结点的____与其到根结点的路径长度之积。
17. 无向图的邻接矩阵是____矩阵。
18. 经过排序的数据, 若其关键字值依从小到大的顺序排列, 则称为____。
19. 假设参与归并排序的两个有序序列长度分别为 n 和 m , 则辅助数组长度是____。
20. B 树都是在____处插入新关键字。



三、解答题:本大题共 5 小题, 每小题 6 分, 共 30 分。

21. 二维数组 $A[9][5]$ 采用行主序方式存储, 每个数据元素占用 4 个存储单元, 若 $A[1][4]$ 的存储地址是 2000, 试计算 $A[7][3]$ 的存储地址。
22. T 是带权无向连通图 $G=(V,E)$ 的最小生成树, 其顶点集合为 V , 边的集合为 E , 试写出克鲁斯卡尔算法的求解过程。
23. 写出下列广义表的表头和表尾。
 - (1) $A=(3,5)$
 - (2) $B=('m',(2,3,'t'))$
 - (3) $C=(A)$
24. 试将数据序列 73,14,21,2,5,34,44,52,61 建成一个最大堆, 并画出堆。
25. 分别计算题 25 图所示三棵二叉查找树在查找成功时的平均查找长度, 假设每个关键字的查找概率相等, 结果保留 1 位小数。



题 25 图

四、算法阅读题:本大题共 4 小题, 每小题 5 分, 共 20 分。

26. 单链表中结点及单链表定义如下:

```
typedef int ELEMType;
```

```
typedef struct node{
```

ELEMType data;

```
struct node *next;
```

```
}LinkNode;
```

```
typedef LinkNode *LinkList;
```

程序为单链表删除 `current` 所指结点的后续结点算法，试在空白处填上适当内容以将算法补充完整。

```
int fun(LinkList head, LinkNode *current, ELEMType *x)
```

```
{   LinkNode *temp;
```

```
if((head==NULL && current==NULL)||current->next==NULL) return 0;
```



```
if(isEmpty(head)==1) return 0; //判空
```

```
*x=current->next->data;
```

```
____(1)____;
```

```
____(2)____;
```

```
____(3)____;
```

```
head->data--;
```

```
return 1;
```

```
}
```

27. 对顶栈的定义如下:

```
#define maxSize 100
```

```
typedef int ELEMTType;
```

```
typedef struct{
```

```
    ELEMTType element[maxSize];
```

```
    int lefttop,righttop;
```

```
}SeqTopStack;
```

程序为对顶栈的出栈算法, 试在空白处填上适当内容以将算法补充完整。

```
int fun(SeqTopStack *mys, ELEMTType *x, int flag)
```

```
{ if((____(1)____&&____(2)____)|| (flag==1&&mys->righttop==maxSize-1)){
```

```
    printf("栈空\n");
```

```
    return 0;
```

```
}
```

```
else {
```

```
    if(flag== -1){
```

```
        ____ (3) ____;
```

```
        return 1;
```

```
    }
```

```
    else{
```

```
        *x=mys->element[++mys->righttop];
```

```
        return 1;
```

```
    }
```

```
}
```

```
}
```

28. 二叉链表中结点及二叉树的定义如下:

```
typedef int ELEMTType;
```

```
typedef struct BNode
```



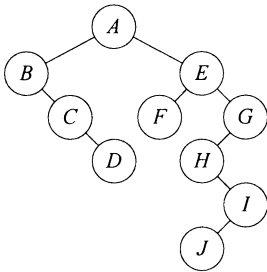
```
{ ELEMType data;
    struct BNode *left, *right;
} BinTNode;
```

```
typedef BinTNode *BTree;
```

(1) 阅读程序，说明 fun 函数的功能。

```
int fun(BTree T){
    if(T==NULL) return 0;
    if((T->left==NULL && T->right!=NULL) || (T->left!=NULL && T->right==NULL))
        return fun(T->left)+fun(T->right)+1;
    else return fun(T->left)+fun(T->right);
}
```

(2) 试将题 28 图所示的二叉树树根传入 fun 函数，求返回结果。



题 28 图

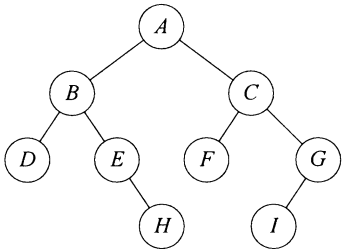
29. 二叉链表中结点及二叉树的定义如下：

```
typedef int ELEMType;
typedef struct BNode
{
    ELEMType data;
    struct BNode *left, *right;
} BinTNode;
```

```
typedef BinTNode *BTree;
```

(1) 阅读程序，说明 fun 函数的功能。

```
void fun(BTree root)
{
    if(root==NULL) return;
    fun(root->left);
    printf("%d\t",root->data);
    fun(root->right);
}
```



题 29 图

(2) 试将题 29 图所示的二叉树树根传入 fun 函数，求输出结果。



五、算法设计题:本题 10 分。

30. 单链表中结点及单链表定义如下:

```
typedef int ELEMTypE;
```

```
typedef struct node{
```

```
    ELEMTypE data;
```

```
    struct node *next;
```

```
}LinkNode;
```

```
typedef LinkNode *LinkList;
```

试编写删除单链表中元素值重复结点的算法。

